# 摘 要

随着业务多样性的发展，光网络向着动态特性、可扩展性等方向迅速发展，然而人们对网络业务的需求远远超过了保障网络生存性技术的更新速度，由于光网络数据传输的高速性，当网络发生故障时，会引起巨大的数据丢失，甚至可能给整个网络带来灾难性后果，因此网络故障的保护和恢复方法需要解决这种由波长容量上升所带来的业务可用性下降问题。在透明光网络中，故障定位是故障保护和恢复的前提条件，当网络故障出现之后，需要根据相应的定位机制来确定故障发生的位置，进而执行保护和恢复操作。因此，准确、快速的链路故障探测与定位方法能够有效降低网络故障所带来的损失，针对光网络中的单链路故障和多链路故障，文中分别给出了一种故障定位的方法。

文中给出了一种基于骑士巡游的单链路故障定位策略。由于网络拓扑、无向图、棋盘三者之间存在一定的联系，根据骑士巡游理论，提出使用棋盘对网络建立故障定位模型；然后提出一种网络节点度数感知的分裂方法，将网络中的所有节点、链路一一映射到相应大小的棋盘上，并且能够反映网络中的节点与链路之间的连接关系，同时构造网络与棋盘的静态映射表；最后依据骑士巡游理论的思想，利用探测信号返回网络中的故障链路信息，通过查找静态映射表定位出故障链路的准确位置。在保证单链路故障完全定位的情况下，对故障定位时间和链路覆盖长度进行了仿真和分析，仿真结果表明在使用较少的网络资源情况下，该策略也能够有效的、快速的定位网络中出现的单个链路故障。

针对网络中的多链路故障，文中给出了一种基于骑士巡游理论的故障定位机制，该机制将网络中的多故障定位过程分为初始化、网元抽象、探测、链路匹配四个阶段。首先根据骑士巡游理论，建立多故障链路定位的网络模型，并利用带有约束条件的网元抽象方法将网络中的所有节点和链路映射到一个带有漏洞的棋盘上；然后在备选探测信号节点周期性发送探测信号，目的节点将收到的有效光路存储在链表中；最后通过链路匹配计算网络中的故障链路。文中分析了多故障定位机制的算法复杂度以及故障定位概率，仿真结果表明在保证业务请求率的情况下可以对网络中并发的多条链路故障进行完全定位，同时能够相对降低故障定位的时间复杂度，从而为解决光网络中的多故障定位技术提供了一种有效的方法。

关键词：光网络，骑士巡游，故障定位，性能监测

# Abstract

With the development of the business diversity, the main character of the optical network in the future is dynamic and scalability. However, the demand for the network business is far more than the rate of the network survivable technology updates. A network failure will cause a huge loss of data, or even could have disastrous consequences for the entire network for the high speed data transmission in the optical network. Consequently, using network fault protection and restoration methods, the problem should be solved, which the business availability is decreasing with the wavelength capacity increasing rapidly. In the transparent optical network, the fault localization is the prerequisite for the fault protection and restoration. When a network failure occurs, the faults location need to be found using the appropriate mechanism, and then the operations of fault protection and restoration is applied. In order to reducing the losses effectively caused by the network failures, the fault localization methods should be accurate and rapid. Therefore, the methods for link failure localization in the optical network are proposed in the paper.

In the paper, based on the knight's tour theory, a novel strategy for the single-link failure localization is proposed. Due to there are some nature of relations between the network topology and chessboard, a fault localization model is established by the chessboard based on knight's tour theory. A degrees aware node splitting method is proposed in the localization strategy. Then, the nodes and links of the network are mapped on the squares and steps of the corresponding chessboard. And the connectivity between the nodes and links is reflected on the chessboard. Meanwhile, the static mapping table is established between the network and chessboard. Subsequently, based on the knight's tour theory, the probing signals are sent periodically to detect the network status in real-time manner. By searching the static mapping table, the single-link failure is located according to returned fault information once the link failure is detected. In the case of achieving unambiguous failure localization of any single link failure in the network, the localization time and cover length of links are analysis. Numerical results show that the single-link failure can be located effectively and rapidly by the proposed strategy, which uses less network resources.

For the multi-links failure localization in the optical network, a mechanism based on

knight's tour theory is presented. And the localization mechanism consists of four phases: initializing, network element abstracting, detecting and matching phases. Firstly, based on knight's tour theory, a multi-fault localization model is established. By using the method of network element abstracting with constraints, the nodes of the network are mapped on the squares of the corresponding chessboard with the holes, whose size is large enough. Secondly, the probing signals are sent periodically at the spare nodes in the network; after a period of time, the effective light-paths which are received at the destination are stored in the list. Therefore, the fault links are computed by matching the links in the list. In the terms of time and space complexities are evaluated for the proposed mechanism in the paper. Meanwhile, the fault localization probability is simulated. And the simulation results show that the proposed fault localization mechanism can be achieving unambiguous failure localization, under the conditions of guaranteeing rate of service request in the network. And the time complexity is relatively lower. Hence, an effective technology is proposed for solving the link failure localization in the network.

**Keywords:** optical network, knight's tour, fault localization, performance monitoring

# 第一章 绪论

光网络作为信息社会主要的通信网络，其承载的业务已由传统的语音业务扩展到数据业务以及多媒体为代表的宽带业务，成为所有通信系统所依赖的主干通信系统。而且为了满足人们对多种类型业务的动态需求，光网络结构已经从点到点的方式向网状组网、智能控制等体现网络动态特性、可扩展性的方向迅速发展，以适应多种业务的突发性和灵活性，从而经济有效地完成多种业务的混合传输。与此同时也为网络管理带来了巨大的挑战，例如网络中若发生光纤断裂、节点失效等故障后，会造成大量的数据丢失，甚至可能给整个网络带来灾难性后果，因此在保障网络的通信质量、可靠性、生存性等网络性能方面，光网络技术需要持续不断地进行更新。

## 1.1 光网络发展过程

随着人们对新业务的不断需求，通信技术得到迅速发展，特别是 IP 业务的迅猛崛起，导致全球信息量呈级数增长，通信业务也由传统单一的电话业务转向高速 IP 数据和以多媒体为代表的宽带业务，对通信网络的带宽和容量提出了越来越高的要求。光纤的巨大潜在带宽和波分复用(Wavelength Division Multiplexing, WDM)技术日益广泛的应用，使光纤通信技术逐渐成为支撑通信传输网络的主流技术，而目前单波长传输速率已经达到 40Gbit/s，进一步提高单波长传输速率的技术仍在不断地更新[1]。

WDM 技术[2]是在一根光纤中同时传输多波长光信号的一项技术，其特点主要有以下几个方面：

(1) 充分利用光纤的巨大带宽资源，在很大程度上解决了传输的带宽问题。

(2) 实现了单根光纤的双向传输，同时 WDM 技术可以将多个波长复用起来在单根光纤中传输，从而节省大量的线路投资。

(3) WDM 技术中的各个波长相互独立，可以同时传输多种不同类型的信号，完成各种业务信号的复用和分离，从而实现多媒体信号的混合传输。

(4) 降低器件的超高速要求，在原有的许多光电器件的响应速度已明显不足的情况下仍可实现大容量传输。

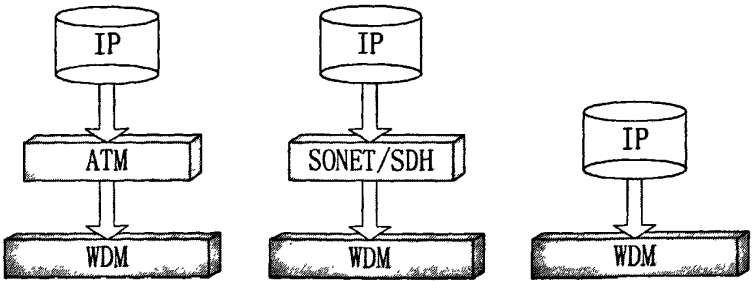
(5) IP 的传送通道，WDM 信道对数据格式是透明的，即与信号的速率和电调制方式无关，在网络扩充和发展中是理想的扩容手段，为引入带宽新业务提供了方便。

(6) 高度的组网灵活性、可靠性、经济性以及多种应用形式，如长途干线的传

输网络、局域网等。

由于 WDM 网络具有高容量、波长路由、透明性、可重构性、兼容性等优越特性，使得 WDM 光传送网为通信传输网络提供了一个大容量、经济、高生存性和灵活性的传输基础设施。WDM 光传送网中的每个波长速率可达 40Gbit/s，单根光纤可传送 160 个以上波长，极大地提高了网络带宽容量；另外，通过光交叉连接和光分叉复用技术的综合使用，WDM 光传送网可以实现光波长信道的动态重构功能，即根据传送网中业务量的变化和需要，动态地调整光路层中的波长资源和光纤路径资源的分配，极大地提高了网络资源的利用率；同时在器件发生失效、线路中断以及节点故障时，为了不影响网络上层业务的传输，可以通过波长信道的重新配置或保护倒换的方式，为发生故障的信道重新寻找路由，使网络迅速实现自愈或恢复[1]。随着人们对网络容量、质量、业务种类等方面提出了更高的要求，通信网络向高速率、高带宽、高服务质量等方向发展，用以满足不断增长的业务需求和对服务质量的要求，而 WDM 光传送网具有的优越特性，使其逐渐成为备受青睐的研究对象。

随着 WDM 技术的日趋成熟，Tbit/s 量级甚至更高传输速率的网络已初步形成，从而对通信网络的容量安排、业务量设计、组网方式等都产生巨大影响，其中为了提高光传送网的交换能力，逐渐提出了 IP over ATM、IP over SONET/SDH、IP over WDM 等光网络结构，如图 1.1 所示为三种网络结构示意图。IP over WDM 是指直接在光层上运行的因特网，综合利用了 IP 技术和 WDM 的光网络技术，合理的在 IP 层与光学层之间实现网络管理、保护恢复、流量工程等优化配置，形成一种简单高效的网络体系结构。采用 IP over WDM 技术，同时省去了 ATM 层和 SDH 层，可以减少网络各层之间的中间冗余部分，从而简化了网络管理，提高了传输效率，降低了额外开销，并且可以与 IP 的非对称业务量特性相匹配，能够充分利用带宽资源，节省网络运营商的成本，从而间接地降低了用户获得通信业务的费用，是一种最直接、最简单、最有效的 IP 网络体系结构，非常适用于大型的 IP 骨干网[3]。




(a) IP over ATM (b) IP over SDH (c) IP over WDM

图 1.1 三种光网络结构

目前整个通信网络正朝着数字化、IP 化、宽带化、智能化和个人化方向飞速发展，而未来网络可能会以软交换为核心，光联网为基础，兼容所有三网技术的开放

体系架构方向发展，而 WDM 技术在实现产业化的同时，将向着更多波长、更高速率、更大容量和更长距离的方向发展。

## 1.2 光网络故障监测技术

网络故障监测技术的目标是通过监控网络的工作状态来实时发现网络中的失效事件和故障情况，为故障定位、故障排除、故障通告和业务恢复提供指示和参考信息，因此，快速、准确的故障监测和定位技术是网络故障保护和恢复机制的前提条件，将直接决定网络生存性。

WDM 技术的迅速发展，单根光纤承载着巨大的数据业务，由于网络传输的高速性，即使一个简单的业务中断，也可能导致社会各个部门发生灾难性问题，因此光性能监测是动态 WDM 网络发展中的一个至关重要的问题。光性能监测技术[4]是利用网络中的监测设备对网络中的关键参数进行实时监测，并及时报告网络的工作状态，其最大目标是最小化监测时间和最大化网络可用性，提高网络资源的利用率。

光性能监测[4]是一种对光信号质量和光信道的物理层监测技术，可以将光性能监测参数分为光信道监测参数和信号质量参数，光性能监测存在于网络中的物理层到网络高层，并且为每一层的数据完整性提供保证，如表 1.1 所示，光信道监测参数提供信道工作状态信息，而信号质量参数被用于评估每条信道上的信号质量。

表 1.1 光性能监测参数分类

| 光信道监测参数 | 信号质量参数 |
|---|---|
| 总功率 | 带内光信噪比（OSNR） |
| 信道功率 | Q 因子、BER、ESNR |
| 信道波长 | 色散 |
| 光谱 OSNR | 比特速率 |

文献[4]-[7]中指出了当前的光性能监测技术不足以支撑未来光交换网络的发展需求，并提出将其中的某些性能监测操作必须向下转移到物理层。由于在光传输网络中各层的物理光网络层和光层监测任务和方法各不相同，虽然将光传输网络分为 IP/ATM/SDH/WDM 等多层，可以提高网络的透明性和数据传输效率，降低网络监测成本，但也使高层更不容易发现物理层上的故障，同时不利于故障的检测和恢复。若可以快速检测和恢复物理层的网络故障，而且高层次的网络能够及时获得更多地物理层信息，就能同时避免数据的丢失和高层错误操作的发生，那么网络的管理将会更加有效[7]。

网络监测[8]可以分为被动监测和主动监测，主动监测包含发送业务到网络中去

抽查网络属性，而被动监测不产生额外的业务量，只是监测通过网络中的某些特定节点的业务，例如统计网络中的分组包，比较复杂的应用是分析、检查分组包的分组头，因此被动监测主要应用于测量网络中的一些特定元件的性能等，例如链路吞吐量、统计包的大小等。但是从应用观点来看，被动监测机制不能保证两个业务传送端点之间的可靠性，使其不能满足端到端的服务质量，而主动监测通常能够获得端到端的数据统计，比如数据吞吐量、损耗和路由可用性等网络特性，另外可以在被选择的节点之间形成业务，使得主动探测在设计探测流时能够提供更好的灵活性，其中探测流包含需要测量的网络属性参数。例如发送探针测量一个路由上的平均延时和损耗，较好的探针测量技术能够识别网络瓶颈和可用带宽或者评估交叉业务量，它的主要缺点是扩散性，有时探针可能会修改路由条件等。

主动监测技术能够有效地解决网络监测应用问题，使用主动探测可以实时监测网络的工作状态，识别产生网络属性紊乱的原因，并诊断网络中故障发生的根本原因，因此主动监测技术的一个有效应用是进行网络故障的定位，其中主动探测机制执行故障定位时包含探针位置的选择、问题探测和问题确定三个步骤。主动探测主要是利用探针策略进行实时观察网络状态，首先发送最小数目的探针进行初始化，然后选择合适的网络位置放置探针，并从选定的探针位置发送探测信号来获取网络信息，这种方式可以减少管理业务、提高精确度和及时诊断。

随着业务需求的迅速发展，使得人们对数据传输质量的要求远远超出网络技术的更新速度，对保证网络的可靠性、安全性、服务质量等方面提出了更高的要求，为了保障网络的稳定性，需要实时监测网络状态，并对潜在的网络危害及时采取保护措施，譬如如何有效进行光信道的性能监测等关键技术还需要进一步的研究。

## 1.3 光网络故障分类

光网络故障监测机制是网络故障定位、恢复能够有效实现的前提条件，监测技术的优劣将直接影响网络性能的稳定性。通过实时监测可为网络的传输质量提供可靠保障，并将相关的网络性能参数发送到网络管理系统，因此当网络发生故障时，能够有效的根据告警信息进行故障定位并及时恢复，从而保证网络传输的有效性。

### 1.3.1 网络故障的分类

网络通信的目的[9]是将信号准确无误的从信源传输到信宿，而信号在传递过程中需要经过多个网络设备，可以通过将监控点放在网络中的不同位置来实时监测网

络的工作状态，如果网络的正常属性发生偏移，监控点就产生告警，那么网络中的某个监控点产生告警必然是前面一个或多个资源发生故障引起的。如图 1.2 所示为网络在双向传输过程中一条割边产生的故障传播模型[10]，其中 M 表示网络中的监测设备，当网络发生故障时，所有经过故障位置的下行监测设备将产生告警信号。由于在透明光网络中缺少光/电/光再生设备，故障可以通过网络中的不同部分进行传播，从而导致大量的冗余告警，增加了处理开销、定位时间和延迟服务恢复，因此为了更加迅速、有效地进行故障探测和定位，必须最小化告警信息数量。

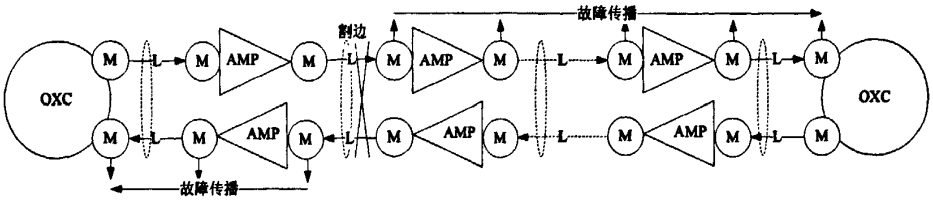



图 1.2 故障传播模型

光网络产生故障的原因很多，包括光纤断裂、能量干扰、色度色散等，根据故障表现形式的不同，可以分为硬故障和软故障[11]。硬故障是指由于光纤断裂、掉电、设备损坏、人为攻击等引起的突然中断通信的事件，虽然硬故障对网络业务的影响非常大，但其出现的形式简单，处理比较方便；软故障是指由于光纤损耗增大、激光器温度漂移、器件老化、波长中心漂移等逐步降低传输质量的事件，说明软故障的发生是故障累积效应，当故障发生时不易被发现，为了防止网络性能的进一步恶化，在意识到网络存在软故障时，必须及时采取有效地保护措施，维持网络性能的稳定性和安全性。若按照故障发生位置的不同可以分为节点故障和链路故障[12]。

(1) 节点故障是发生在光网络的节点内，光发射机、光接收机或相关设备等由于器件本身的老化以及人为因素的影响所导致的故障。光网络节点中一般包含有源器件和无源器件，其中固定封装的无源器件（如光隔离器、环形器、耦合器、光纤光栅）性能比较稳定，发生故障的概率很小。因此，节点故障主要是由各种有源器件所产生的故障，各种有源器件所引起的故障主要包括发射机故障（激光器或驱动电路损坏）、接收机故障（光检测器或接收电路损坏）、光放大器故障（泵浦激光器损坏等）、OXC(Optical Cross-connection, OXC)故障、波长转换器故障（半导体光放大器 SOA 或基于饱和吸收的分布 Bragg 反射激光器损坏）。

(2) 链路故障是指由于自然因素和人为因素的影响，导致光纤断裂或性能劣化，造成通信质量的下降或网络业务的中断，一般的网络故障是指链路故障。

基于告警的故障探测和定位机制被广泛用于解决光网络中的网络监测和故障定位问题，由于网络故障和告警相互依存，告警是网络故障产生的必然结果，因此光网络发生故障时，监测设备就向网络管理系统发送一些告警信息，由此可见，告警是故障的外在表现形式。在光网络中，由于不同类型的监测设备对网络中的同一故

障产生不同的告警信息，使得网络管理系统接收到大量的冗余告警信息，因此为了更加有效地控制告警产生的范围和减少告警数量，提高网络服务恢复的速度，需要高效地识别和处理网络中产生的告警信息，从而找出故障发生的根本原因。

### 1.3.2 故障定位技术的研究现状

光性能监测的目的是为了找出网络运行过程中存在的性能紊乱，并将告警信息发送到网络管理系统，从而为故障定位提供可靠的依据。目前，基于监测设备提出的光网络故障监测和定位技术的研究相对较多，为了降低网络监测成本，最小化监测器数目、降低监测周期和带宽开销等网络监测和定位技术被不断地优化。

关于故障监测和定位技术的文献中均对网络故障产生的原因进行了分析，并且根据相应的算法或协议机制进行故障探测和定位，同时也阐述了故障定位机制在网络管理中的重要作用，即故障恢复和保护的前提，并提出为了更好的利用网络资源，保障网络性能，必须找出关于定位网络故障的有效方法。

文献[13]分析了透明光网络中故障发生的类型，如何探测故障和故障影响其它光路的方式等一般的故障管理问题，并提出在选择合适的监测位置后，只需对功率、波长偏移、带内干扰、带外干扰进行分析的故障定位算法。

利用波分复用技术，文献[14]提出故障探测和定位方案来管理光网络中的多故障，该方案首先使用监测设备进行故障或攻击探测，然后通过发送和接收探测信号来分析网络故障或攻击的潜在位置，并运行定位算法对网络故障或攻击进行定位。

文献[15]提出了有限周边矢量匹配(Limited-perimeter Vector Matching, LVM)协议来定位全光网络中出现的单链路故障，LVM 协议是在分布式控制和管理机制的基础上进行构造的一种新颖的故障定位协议，在建立光路的中间节点跳过任何光功率监测器或频谱分析等监测设备，并将故障定位区域限制在一个很小的范围内进行，从而提高了网络的透明性和降低故障定位时间。在文献[16]-[18]中通过增加优化业务分布等限制条件，利用整数线性规划问题对 LVM 协议性能进行优化，尽量达到故障定位概率最大化和定位时间最小化。

文献[19][20]提出了一种预配置监测波长回环连接的圈覆盖算法，在全光网络中使用波长作为监测信道进行网络故障的探测和路径性能监测，通过圈覆盖发现算法将网络拓扑分解成圈的集合，并进行圈上故障链路的定位。

通过移除圈约束条件，文献[21]提出了将光发送设备和光接收设备放置在不同的节点位置的监测迹算法，解决了圈内故障定位问题，并且能够对网络中的单个链路故障实现完全定位。文献[22]提出避免告警洪泛和移除电信号机制的监测迹算法，沿着每条监测迹的多个监测节点，通过窃听光监测信号来监测每条监测迹的工作状

态，同时共享这些状态信息，并根据局部有效的告警比特定位光网络中的单个链路故障，是一种利用监测迹方法进行网络故障定位的优化算法。

文献[23]提出采用监测树算法定位二维格状网络中的单链路故障，并以巧克力棒图为模型，利用经典纠错码构造告警代码的方式定位网络中的单个链路故障。

针对网络中任何单链路引起的故障情况，文献[24]提出了基于监测圈和监测路径的故障定位方法，通过分析网络拓扑状态，选定网络中的某些特定节点作为监测节点，利用监测圈和监测路径的组合对网络中的各条链路进行性能监测，然后利用告警信息对存在的故障链路进行完全定位。

文献[10]首先对现存的基于建立监测圈和路径的监测技术进行了分析，并指出现存网络监测技术的不足之处，同时给出一种集中控制和分等级分布式监测模型的主动激活监测器的故障定位算法。

文献[25]中指出大部分故障推理算法可用二部图来描述故障特征和故障原因之间的关系，在此基础上，文中利用故障症状行为模型将主动调查行为融合到被动故障原因进程中，提出了一种新颖的主动整合故障原因的故障定位技术。

文献[26]提出基于多路供应的故障定位方案，文中使用状态转移模型识别多层的网络故障，然后根据不同的算法定位网络故障，包括单链路故障、多链路故障和灾难性故障。

文献[27]提出将高层故障识别转化成较低层故障产生的根本原因进行定位，并给出了一种风险模型进行网络故障的识别和定位。

文献[28]考虑不同网络场景的共享风险链路组，使用监测迹和监测树算法对网络中的单个链路故障进行完全定位。

文献[29]-[40]等其它文献中也提出了解决网络故障定位的方法，部分文献对现有的一些故障定位算法进行了性能优化，从而进一步降低网络故障监测和定位成本。

## 1.4 本文的主要工作和内容安排

随着光纤通信技术的迅速发展，光传输网络向着远距离传输、大容量通信的方向发展，使得多媒体等宽带数据业务也日益兴起，然而人们对网络业务的需求远远超出了网络技术的更新速度，因而保障网络数据传输的可靠性和服务质量的技术也在广泛的研究当中。由于光网络数据传输的高速性，当网络发生故障时，将会造成巨大的数据丢失，并可能对社会某些部门产生灾难性影响，因此提高网络的生存性，尽可能降低因网络故障而造成的业务损失和社会影响仍是一个很严峻的问题。

网络生存性[41]是指网络发生故障时，恢复受损的业务，使网络维持在一个可以

接受的业务水平的能力，网络生存性的实现可以分为故障监测、故障定位、故障通知和故障恢复。网络的故障恢复分为保护和恢复两种，而所有的故障恢复机制都需要依次完成故障监测、故障定位、故障通告和故障恢复操作，因而故障监测和定位是网络各项生存性技术的基础，如果网络发生了故障，只有快速的监测到故障，才能进行后续的故障定位、故障通知等恢复动作。

本文主要研究了光网络中的链路故障定位问题，为了降低网络的故障定位时间，减少网络资源的开销，并达到对故障的完全定位，提出了基于骑士巡游理论定位网络中的链路故障的方法，本文内容安排如下。

第一章，绪论。主要介绍光网络的发展，随着 WDM 技术的日益成熟，未来光网络逐渐成为网络业务传输的载体，为了保证网络传输的有效性和服务质量，相关技术被广泛的研究，因此对光性能监测和网络故障等方面的内容进行了简单的介绍。

第二章，故障定位机制。网络故障监测和定位是故障保护和恢复的前提，是网络生存性技术的基础，因此通过阅读大量的参考文献，阐述了光性能监测技术面临的挑战，并介绍了几种故障监测和定位技术。为了降低网络成本，在快速、准确定位故障链路的同时应最小化监测代价，在所阅读的参考文献中也对文中提到的几种故障监测和定位技术进行了性能优化，从而进一步减少故障定位时间和降低网络资源的开销，而其它较多的故障监测和定位技术并没有一一列举。

第三章，基于骑士巡游的单链路故障定位策略。准确、快速的链路故障探测与定位方法能够有效降低网络故障所带来的数据丢失和经济损失，全光交换网络的链路故障定位方法更需要具有快速性，同时能够有效降低资源开销，在本章中提出了一种基于骑士巡游理论的光网络中单链路故障定位策略，并介绍了该策略定位网络链路故障的过程，通过对不同的网络拓扑进行验证，仿真结果表明骑士巡游理论能够解决网络中的链路故障定位问题，并对故障定位时间和链路覆盖长度进行了分析。

第四章，基于骑士巡游的多链路故障定位机制。针对光网络中的单故障定位技术，现有文献中均对网络中的故障进行了分析，并根据相应的算法机制进行故障探测和定位，同时也阐述了故障定位机制在网络管理中的重要作用。然而现有文献中对网络多故障问题没有较多的研究，并且单故障定位方法并不能为网络中的多故障给出快速、有效地判断，因此在本章中，根据骑士巡游理论提出了一种多链路故障定位机制。仿真结果表明该定位机制能够有效地定位网络中并发的多条故障链路，同时计算了该定位机制的时间和空间复杂度。

第五章，总结。对论文的研究工作以及取得的成果进行了总结，并分析了骑士巡游理论在光网络故障管理中可能的应用，同时对未来的研究工作进行了计划。

# 第二章 故障定位机制

随着人们对网络的依赖性越来越大，对网络性能以及服务质量的要求也越来越高，使得网络生存性的地位日益提高，因此关于网络生存性的研究一直备受重视。网络的生存性[41]是指网络发生故障时，恢复受损的业务，使网络维持在一个可以接受的业务水平的能力，以降低因故障而造成的业务损失和社会影响，它是网络完整性的一部分。网络生存性的实现可以分为故障监测、故障定位、故障通知和故障恢复，而故障监测和定位是网络保护和恢复的前提条件，因此快速有效的故障定位技术是保障网络性能稳定的关键技术之一。

## 2.1 光性能监测技术面临的挑战

光网络的快速发展，使得其具有大容量和高速的传输速率，逐渐成为多媒体等宽带业务的主要传输载体，与此同时对光网络管理提出了新的挑战。网络业务在高速的传输前提下，一个短时间的服务中断就可能会造成大量的数据丢失，因此高速透明传输网络的生存性管理是相当重要的。另外随着光网络的交换方式由静态可重构向动态可重构的方向发展，光网络的快速动态化、可扩展性等特性，使得整个传输网络更容易受到网络故障、错误配置、信号质量恶化等因素的影响。有效地光网络性能监测技术和故障定位技术才能使光网络的生存性得到保障，因此，网络的性能监测技术是保障网络性能稳定的关键技术之一。

光性能监测技术[4][5]的最大目标是最小化监测时间和最大化网络的可用性，通过减少系统不必要的操作来提高网络资源的利用率，其中光性能监测的潜在应用包括网络元件控制，链路的配置、控制和优化，基于物理特性约束的链路选择、故障和损伤发现、故障探测和定位、针对信号质量下降激活告警等光网络参数特性的应用。随着光网络的发展，透明化、智能化、高速长距离化逐渐成为其发展趋势，为了保证光网络管理的可靠性，光性能监测已经发展成为一个重要的研究课题，面对新的网络发展趋势，光性能监测技术也面临着新的挑战[5]。

(1) 光性能监测技术被广泛的提出，但是没有足够完美的技术来解决光网络中众多参数的监测问题。在未来光网络中，多重数据速率和调制方式将同时存在，而光性能监测必须对它们进行透明监测，同时还应该在低损耗、高精确度、快速响应、广泛的参数覆盖等参数性能监测方面进行深入的技术研究。

(2) 在电域中的性能监测，所有参数特性可以在一个单一的设备中进行计算，

相比之下，光性能监测技术中，一种监测技术只能被用于监测一个特定的参数特性，如果有较多的参数需要被监测，则增加了光性能的监测成本。

(3) 如何有效的利用光性能监测参数，例如如何利用光性能监测参数来探测网络故障，并分析故障产生的根本原因，然后准确地判断网络故障发生的位置。

因此，网络的高速化趋势要求光网络必须具有高可靠性和生存性，这样才能保证在光网络中出现故障的时候，能够尽量减少数据的丢失。高质量的光网络监测技术是光网络可靠性得以保障的前提条件，从而要求更先进的光网络监测技术必须具备准确性以及实时性。

## 2.2 光网络的故障定位技术

在光网络中，虽然故障发生的概率不是很大，但是一旦网络发生故障将会造成巨大的数据丢失和经济损失，甚至对整个网络产生灾难性影响，例如单个网络故障，比如光纤断裂、节点失效等故障，就会造成业务数据的丢失，极大地降低了网络的服务质量，甚至造成网络服务中断。因此为了确保网络的服务质量，网络必须包含有效的故障保护和恢复机制，以提供针对各种网络故障的高水平的服务生存性。

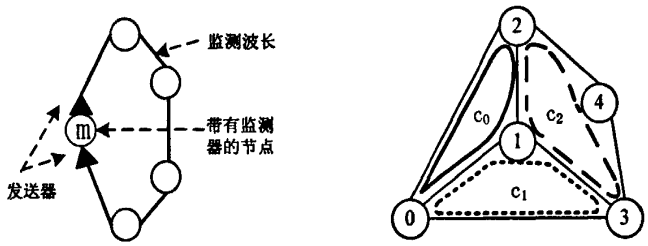
网络的故障恢复分为保护和恢复两种，保护是利用节点之间预先分配的带宽资源迅速恢复受损业务的传输，而恢复则是在故障发生以后，故障路由利用节点之间的可用资源动态地进行重路由来代替，而不需要预先进行带宽资源的预留。所有的故障恢复机制都需要依次完成故障监测、故障定位、故障通告和故障恢复操作，因而故障监测和故障定位是网络各项生存性技术的基础，如果网络发生了故障，只有快速的监测到故障状态，才能进行后续的故障定位、故障通知等恢复操作。网络故障发生后，故障的具体位置是由光网络中的节点进行确定，因此要求故障定位机制必须准确且快速的确定故障位置，从而进行故障链路的恢复或保护，尽可能减少由于网络故障而造成的网络资源的浪费、业务的大量丢失以及经济上的损失。

目前，针对光网络故障定位机制的研究相对较多，在所阅读的文献中，大部分提出的故障定位技术是基于监测设备来探测网络性能的变化，通过使用有效的控制手段或核心算法来实时监测网络性能并进行故障的判断和定位。下面将针对目前比较经典的几种故障定位算法或机制进行了简单介绍。

### 2.2.1 圈覆盖定位算法

圈覆盖定位算法[19][20]是由 Zeng Hongqing 等人提出的比较经典的故障定位算法

之一，该算法的主要思想是把网络分解为圈的集合，使得网络上的每个节点和每条链路至少被集合中的一个圈所覆盖，并分配一个监测器模块到每个圈中。一个网络监测器被分给每个圈中的一个节点，并通过收发器发送和接收监测波长，则在该圈中形成一个回环监测信道，因此，这个带有监测器和监测信道的圈被称为监测圈(monitoring cycle, m-cycle)。如图 2.1 所示，在带有监测器的节点 m 处发送一个监测波长，通过网络中的某些链路后返回到该节点；图 2.1(b)为使用监测圈来定位故障链路的网络拓扑图，该网络被分解成的监测圈的集合 C={c0,c1,c2}，其中每个监测圈的路径分别为：c0：0-1-2-0，c1：0-1-3-0，c2：1-2-4-3-1。由于监测圈上的网络故障引发的告警，可以通过监测圈的告警集合来确定网络中的哪一条链路发生故障，如果一个网络告警只在某些监测圈中发生，则说明故障链路为这些监测圈的共享链路。表 2.1 为该网络拓扑使用圈覆盖算法对网络中每条可能发生故障的链路的告警代码，由于链路(2,4)和(3,4)拥有相同的告警代码，故当只有监测圈 c2 发生故障告警时，不能够确定这两条中的哪一条链路发生故障。




(a) 监测圈 (b) 监测圈定位链路故障

图 2.1 监测圈故障定位模型

表 2.1 监测圈的故障告警代码表

| Link | c2 | c1 | c0 | Decimal |
|---|---|---|---|---|
| (0,1) | 0 | 1 | 1 | 3 |
| (0,2) | 0 | 0 | 1 | 1 |
| (0,3) | 0 | 1 | 0 | 2 |
| (1,2) | 1 | 0 | 1 | 5 |
| (1,3) | 1 | 1 | 0 | 6 |
| (2,4) | 1 | 0 | 0 | 4 |
| (3,4) | 1 | 0 | 0 | 4 |

目前，使用构造圈进行定位网络中的单个链路故障的圈覆盖算法中，比较经典的故障定位算法分别有启发式深度优先搜索(Heuristic Depth First Searching, HDFS)算法、最短路径欧拉匹配(Shortest Path Eulerian Matching, SPEM)算法和启发式生成树(Heuristic Spanning Tree, HST)算法。

启发式深度优先搜索算法[19]的主要思想如下，假设在一个无向图 $G(V,E)$ 中，

由任意节点$n \in V$开始，利用深度优先搜索算法遍历$G$中的所有链路，被遍历过的部分记作$G'(V',E')$。当从节点$x$到节点$y$的链路$e$被访问时，如果$y \in V$，则一定存在一条路径$p(y,\cdots,x) \in G$，因此这条路径和链路$e(x,y)$形成了一个圈。

最短路径欧拉匹配算法[19]思想是使用欧拉匹配算法先把无桥连通图扩展为欧拉图。因为对于每个欧拉图，都存在一个欧拉圈，它覆盖图中的所有链路一次且仅一次，如果通过遍历欧拉图中的链路，直到一个节点被再次遍历，就形成了一个圈，那么从欧拉图中移去该圈包含的所有链路；然后遍历剩余部分，直到所有的链路都被移去。根据上述方法，欧拉图被分解成了圈的集合，在把无桥图扩展成欧拉图的过程中添加的链路，相当于原图中的链路被遍历了两次。

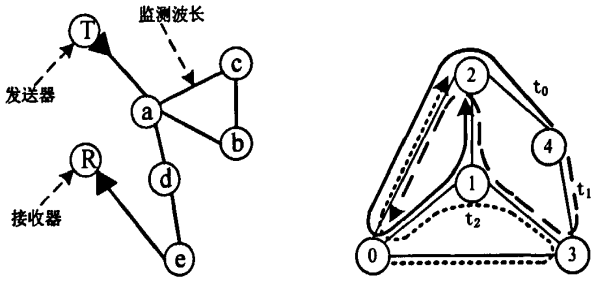
启发式生成树算法[20]主要是使用广度优先搜索的方法构建生成树，假设在无向图$G(V,E)$中，首先选择图中连通度最大的节点作为生成树$T$的根节点，并将与根节点相连的链路放入生成树$T$中，同时将与这些链路相连的网络节点放入$T$中；然后更新$T$中每个节点与不在$T$中节点之间的连通度，选择$T$中度数最大的节点作为根节点，并将与该根节点相连的且不在$T$中的链路和节点放入到$T$中。根据上面的方法将网络中的所有节点放入到生成树$T$中，并将网络中不在$T$中的链路作为弦，最后为所有的弦构造圈，而这些圈就是定位网络中的链路故障所需要的监测圈。

圈覆盖算法在定位网络中的链路故障的同时需要考虑故障定位率、波长开销和监测器数目三个方面，使用以上三种圈覆盖算法对网络中的单个链路故障进行定位，在相同的仿真环境下，HST 算法比 HDFS 算法、SPEM 算法具有更好的成本节约，而这三种圈覆盖定位算法均比基于信道监测和基于链路监测的方案耗费更少的网络成本，其仿真结果在文献[20]中有详细的介绍。

### 2.2.2 监测迹定位算法

当监测圈发生故障告警后，圈覆盖算法在定位网络中连通度为 2 的节点时，存在无法确定与该节点相连的哪一条链路发生故障，因此 Wu Bin 等人针对此问题，提出了一种新的故障监测和定位方案。由于循环监测结构限制了监测器资源分配的灵活性，因此通过移除圈约束条件，形成由简单圈、非简单圈和开迹构成的监测迹(monitoring trail, m-trail)算法[21]。一条监测迹可以多次通过同一个节点，但是一条链路只能通过一次，如图 2.2(a)所示，将光发送器 T 和光接收器 R 分开放置在网络中的节点上，监测波长信道为 T-a-b-c-a-d-e-R 或 T-a-c-b-a-d-e-R 均可，其中在同一个监测波长下的不同预交叉连接方式并不影响监测结果，这种预交叉连接路径称为监测迹。图 2.2(b)为使用 m-trail 算法定位链路故障的网络拓扑图，为了区别网络中的所有链路故障，图中使用了三条监测迹{t0,t1,t2}来进行描述，每条 m-trail 分别为：

t0：4-2-0-1-2，t1：4-3-1-2-0，t2：0-3-1-0-2。由于网络中的所有链路需要通过 m-trail 监测其性能状态，故通过 m-trail 的告警代码集合，可以判断网络中故障链路的准确位置，表 2.2 为 m-trail 的告警代码表，从表中可以看出网络拓扑中的每条链路都有唯一的告警代码值，因此 m-trail 算法可以对网络中的链路故障进行完全定位。




(a) 监测迹 (b) 监测迹定位链路故障

图 2.2 监测迹定位故障模型

表 2.2 监测迹的故障告警代码表

| Link | t2 | t1 | t0 | Decimal |
|---|---|---|---|---|
| (0,1) | 1 | 0 | 1 | 5 |
| (0,2) | 1 | 1 | 1 | 7 |
| (0,3) | 1 | 0 | 0 | 4 |
| (1,2) | 0 | 1 | 1 | 3 |
| (1,3) | 1 | 1 | 0 | 6 |
| (2,4) | 0 | 0 | 1 | 1 |
| (3,4) | 0 | 1 | 0 | 2 |

使用 m-trail 算法定位链路故障的监测成本为监测器成本和带宽成本之和，带宽成本可以通过计算所有 m-trail 的链路覆盖长度来衡量，故为了最小化监测成本，需要最大程度地减少监测器数目和链路覆盖长度。文献[21]、[22]、[29]等文献中针对最小化监测成本的问题进行了详细的分析和介绍，并对算法性能进行了优化。

### 2.2.3 有限周边矢量匹配协议定位网络故障

有限周边矢量匹配(Limited-perimeter Vector Matching, LVM)协议[15]是 H.T. Mouftah 等作者提出用于解决光网络中的链路故障定位问题的一种协议机制，它是一种分布式故障管理机制，具有高可测量性和速度性，并且无成本，但是以增加计算复杂度为代价的单链路故障定位协议。

LVM 协议将定位网络中的单个链路故障分为洪泛、暂停、多播、匹配和总结五个阶段，LVM 协议主要是将网络中发生链路故障的可能位置限制在一个较小的区域

中，从而降低故障定位的时间和空间复杂度，同时，该协议假设网络中除 sink 节点外，光路的所有中间节点都没有探测光功率损耗或光信号质量下降等监测功能，因此 LVM 协议在光路传输过程中跳过功率监测、频谱分析等检测光信号性能的监测设备，从而最大程度地体现了光传输的透明性。当网络发生故障时，通过一组连接 sink 节点的光路检测数据业务，可以探测到网络中的信号损耗和高信噪比，而最接近故障链路的 sink 节点能够比其它的 sink 节点更早地发现故障，则这个 sink 节点被称为潜在执行 sink (Potential Executive Sink, PES)节点。如果相关的光交叉连接 OXC 丢失数据的路径不止一条，则选择最短的光路作为 PES 节点，LVM 协议的故障定位进程如下所示：

(1) 一旦网络中的某条链路发生故障，那么通过该条链路的所有光路都将会受到影响，因此每条被干扰光路的 sink 节点都会探测到信号损耗和其它故障特征。

(2) sink 节点探测到故障后，选择被影响的光路中最短光路的 sink 节点作为 PES，同时发送一个告警信息到它域内的所有节点，其中，告警信息包含了节点编号和最短光路的长度。

(3) 经过一个等待周期后，确保每个 PES 节点都收到来自其它 PES 节点发送的告警信息副本，然后比较每个 PES 告警信息中的路径长度，选择最短路径告警信息的 PES 作为主动执行 sink 节点(Executive Sink, ES)，若包含最短路径的告警信息中存在多个 PES，则选择节点编号最小的 PES 作为 ES 节点。

(4) ES 节点根据被影响的光路创建被影响链路矢量(Affected Link Vector, ALV)，将包含所有的 ALV 链路和它们的邻居节点的区域定义为有限周边，然后将 ALV 信息多播到有限周边的所有节点中。

(5) 在有限周边中的每个 sink 节点收到 ALV 后，将创建自己的链路矢量，并与 ALV 作比较。如果该 sink 节点的链路矢量与 ALV 有相同的链路，则发送一个二元矢量到 ES 节点。二元矢量包含链路匹配矢量和相应光路的工作状态，如果一条光路无故障，它的工作状态置为 1，否则光路工作状态置为 0；其中故障链路的匹配矢量为 1，否则为 0。对于无故障光路，若 ALV 中的某些链路在 sink 节点的链路矢量中被发现，则将这些链路的匹配矢量置为 0，将 ALV 中其它链路的匹配矢量置为 1；对于故障光路，若 ALV 中的链路包含在 sink 节点的链路矢量中，则将该条链路的匹配矢量置为 1，否则将链路的匹配矢量置为 0。

(6) ES 节点接收有限周边中的所有 sink 节点发送的二元矢量，然后在收集到的二元矢量中执行逻辑与操作，从而确定故障链路的位置。如果所有的二元矢量执行逻辑与后，链路矢量为 1 的链路被确定为故障链路，若二元矢量逻辑与结果中含有多个 1，ES 节点将通过扩展有限周边的区域进行第二次寻找故障链路。

(7) 一旦确定故障链路的位置，ES 节点广播一个故障通知信息，将故障链路发

生的位置发送给网络中的所有节点。

文献[16]-[18]中对 LVM 协议定位网络中单个链路故障的情况进行了详细的介绍，同时通过限制网络业务分布等约束条件，对 LVM 协议定位链路故障问题进行了优化，使其在应用过程中具有更好的性能。文献[39]中使用 LVM 协议定位多域中的单链路故障，并将故障定位分为域间定位阶段、带有域间光路影响的域内定位阶段和没有域间光路影响的域内定位阶段，同时对其时间复杂度进行了分析，仿真结果表明LVM协议定位多域中的单链路故障的平均时间延迟低于OSPF(Open Shortest Path First, OSPF)协议的故障定位时间 40ms。

针对全光网络中的多链路故障情况，文献[40]中对 LVM 协议进行了改进，并提出了并行有限周边矢量匹配(Parallel Limited-perimeter Vector Matching，P-LVM)协议来定位网络中出现的多条链路故障。当全光网络中并发多条链路故障时，P-LVM 协议首先在识别故障后尝试分离每个链路故障到一个小的有限周边区域中，然后以分布式的方法对所有的故障链路进行同时定位。使用 P-LVM 协议可以有效的定位光网络中同时发生的多条故障链路或不在同一时刻发生的多条故障链路，该协议的空间复杂度比较高，而且不能对网络中的所有故障链路进行准确定位，存在一定的误定位链路和漏定位链路。

### 2.2.4 基于风险模型的故障定位方法

虽然将光传输网络分为 IP/ATM/SDH/WDM 等多层，可以提高网络的透明性和数据传输效率，降低网络监测成本，但也使高层更不容易发现物理层上的故障，同时不利于故障的检测和恢复。若可以快速检测和恢复物理层的网络故障，而且高层次的网络能够及时获得更多的物理层信息，就能同时避免数据的丢失和高层错误操作的发生，那么将会使网络的管理更加有效，因此，对于物理层故障的检测和隔离显得越来越重要[7]。利用高层监测技术来探测低层或交叉层故障，但是缺少工具来精确定位一个被探测到的故障发生的位置，文献[27]中利用风险模型的有效性，将高层的故障通知转换成低层发生故障的根本原因，使用风险模型孤立光器件和监测到的物理层故障。

文献[27]中提出的风险模型定位网络故障的方法包含三个方面：故障探测和风险模型、核心定位算法、简单改动来管理特殊域的缺陷。将故障特征和故障原因用二部图来表示，使用空间相关和奥卡姆剃刀定律（简单有效原理，如无必要，勿增实体）对网络故障发生的位置进行判断。

故障管理的必要问题是探测定位，减轻和最终纠正网络性能下降的任何情况，为了帮助故障管理，很多网络元件被设计成具有持续监测一定属性和发出故障告警

的功能，并且不同类型的监测设备对同一故障可以产生不同的告警信息，但是诊断问题的挑战是告警只能表明一个网络元件观察到网络正常属性的偏移，而实际故障可能发生在网络中的任何地方。在综合一组潜在的相关告警下，根据高层风险模型，包含映射观察器件和潜在原因之间的依赖关系，构造一个依赖网络拓扑的风险模型，并将网络中可能的故障和产生故障原因之间的关系用二部图来表示，最后从故障探测系统收集故障特征，将故障特征和风险模型作为条件输入文献中给出的核心定位算法，并输出网络中一组可能故障的假设，进行实时和离线故障分析，从而确定故障发生的根本原因和故障位置。

## 2.3 本章小结

本章简单介绍了光网络中光性能监测技术的研究现状和几种典型的网络故障定位机制，目前对光网络故障定位机制的研究相对较多，大部分文献中提出了基于监测设备来探测网络性能并进行故障定位的方法，并且能够准确地、完全地进行故障定位。m-cycle、$m^2$-cycle、m-trail、监测树、LVM 协议、监测圈和监测路径、风险模型等故障定位算法均能够对网络进行性能监测和故障定位，为了降低网络监测成本，在快速地、准确定位故障链路的同时应最小化监测代价，故在算法优化的同时需要考虑：(1)监测器数目最少化，即硬件监测成本最小化，提高网络监测模型的经济性；(2)平均圈长最小化，即降低故障监测周期，提高监测模型的实时性；(3)平均波长开销最小化，即降低故障监测过程中所带来的带宽开销；(4)故障定位度最小化，即降低故障定位度使它接近于 1，从而达到完全定位。

文献[13]-[40]中主要是针对光网络中的单链路故障进行定位，这些文献均对网络中的链路故障进行了分析，并根据相应的算法或机制进行故障探测和定位，同时也阐述了故障定位机制在网络管理中的重要作用，是故障恢复和保护的前提，并提出为了更好的利用网络资源，降低监测成本，必须找出关于定位网络故障的有效方法。而现有文献中，针对网络的多故障定位问题没有较多的研究，当网络发生多故障时，这些单故障定位方法并不能对网络中的多故障给出快速、有效地判断，因此，在研究网络的多故障定位机制方面任重道远。

# 第三章 基于骑士巡游的单链路故障定位策略

随着透明光网络的迅速发展，波分复用技术被广泛使用，光纤速率能够达到40Gb/s或者更高，当网络发生故障时，会引起巨大的数据丢失，甚至对整个网络产生灾难性影响，因此网络故障的保护和恢复方法需要解决这种由波长容量上升所带来的业务可用性下降问题。在透明光网络中，故障定位是故障保护和恢复的前提条件，当网络故障出现之后，需要根据相应的定位机制来确定故障发生的位置，进而执行保护和恢复操作。因此，准确、快速的链路故障探测与定位方法能够有效地降低因网络故障所带来的经济损失。

全光交换网络的链路故障定位方法需要具有快速性，同时应能够有效降低网络资源的开销，根据骑士巡游理论，本章提出一种光网络中的单链路故障定位策略，该策略首先建立关于故障定位的网络模型，然后对网络进行节点度数感知分裂，并根据分裂后的网络节点数构造相应大小的$m\times n$棋盘，同时进行网络与棋盘的映射；最后依据骑士巡游理论的思想，利用探测信号定位网络中可能出现的单个链路故障。

## 3.1 故障定位的网络模型

一般情况下，网络拓扑可以通过无向图来进行描述，网络中的节点用无向图的顶点表示，网络中的链路用无向图的边表示。用图$G(V,E)$表示一个包含$V$个顶点和$E$条边的无向图，若一个网络拓扑通过无向图$G(V,E)$进行描述，则$V$为网络中的节点个数，$E$为网络中的链路数，因此通过构造邻接矩阵，可以反映网络中各个顶点与各条链路之间的连接关系。若将棋盘上的方格作为无向图的顶点，方格之间的连接关系作为无向图的边，那么棋盘上的方格及其方格之间的连接关系可以看作是一个特殊的无向图。因此，通过一定的方法，可以将网络拓扑中的所有节点和链路之间的连接关系反映到棋盘上，并且不改变网络的连通性。根据骑士巡游理论，本章将网络中的所有节点和链路分别用棋盘的方格以及方格之间的合法马步来表示，通过骑士巡游行走路线来实时监测网络中的链路状态，从而解决网络中的单个链路故障定位问题。

骑士巡游理论[42]为马遍历棋盘问题，骑士（马）的遍历是为马能走遍棋盘的每一个方格且每格恰好经过一次，还可以进一步要求马遍历整个棋盘后恰能回到出发点，这样的一条路径称为马的回路。一般的解决方法是把马的遍历问题转化成图论问题进行分析并解决，可以把棋盘的格子作为图的顶点，相互之间可以经马步到达

的格子之间以边相连，因此，马步图是将棋盘中的方格作为顶点的集合，骑士的合法马步作为边的集合构成的一个简单无向图，进而解决骑士遍历棋盘问题[42]-[44]。

假设骑士巡游遍历棋盘的大小为$m\times n$，实际上棋盘的方格可以与平面直角坐标系上的整点集合相对应[45]，并且可以使用无向图$G(V,E)$来描述骑士巡游路线图，若用节点$V=\{(x,y)\mid 1\le x\le m,1\le y\le n\}$来表示骑士的当前位置，故根据骑士巡游理论，$m\times n$棋盘上骑士的合法线路为$E=\{(x_1,y_1),(x_2,y_2)\mid(|x_1-x_2|,|y_1-y_2|)=(1,2)\}$或者$E=\{(x_1,y_1),(x_2,y_2)\mid(|x_1-x_2|,|y_1-y_2|)=(2,1)\}$。因此若用$(x,y)$表示骑士的当前位置，则相对骑士在棋盘上的当前位置，骑士下一步要走的行、列上的相对坐标分别为：$x=\{-2,-1,1,2,2,1,-1,-2\}$和$y=\{1,2,2,1,-1,-2,-2,-1\}$。因此若位于$m\times n$棋盘上第$i$行$j$列格记为$a(i,j)$，并令骑士的当前位置为$(i,j)$，因为骑士的合法棋步是$(i\pm1,j\pm2)$或$(i\pm2,j\pm1)$，因此当$a_0=i+j$时，马的下一个合法位置是$a_1=i+j\pm3$或$a_1=i+j\pm1$。如图 3.1 所示为棋盘上基于骑士当前位置方格 H 的合法马步示意图，其中标示数字 0-7 的方格为 H 下一步合法马步的位置，同时说明在棋盘上一个方格最多有八个合法马步。若骑士的当前位置是棋盘上四个角点之一的方格，那么该骑士最多只有两个合法的马步。

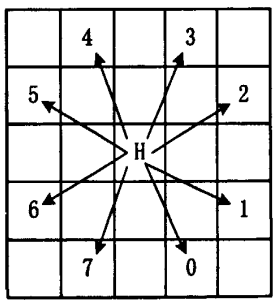
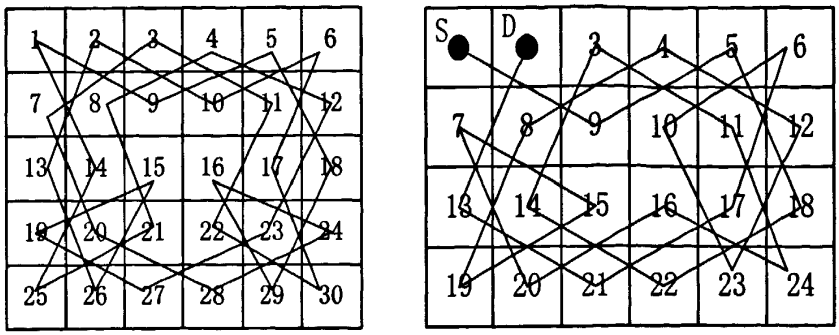



图 3.1 骑士当前位置 H 的合法马步

根据文献[43]和[46]中解决骑士巡游理论的算法思想，可以在任意$m\times n$棋盘上寻找出一条骑士巡游路径。如图 3.2(a)所示为在 5×6 棋盘上寻找的一条骑士巡游路径，该条路径从棋盘的一个角点(方格 1、6、25、30 为该棋盘的四个角点)出发，并且在遍历整个棋盘后返回到该角点的哈密顿回路。本章主要是利用骑士巡游理论在$m\times n$棋盘上寻找一条哈密顿路来构造故障定位的模型，因此降低了寻找骑士巡游路径的算法复杂度，如图 3.2(b)所示为一条 4×6 棋盘上的骑士巡游路径，从棋盘的一个角点(方格 S)出发，遍历棋盘上的各个方格后返回到该角点的相邻的方格 D 中。

根据网络拓扑属性，可以将网络中的节点和链路信息反映到一定大小的棋盘上，并且利用骑士巡游理论，将网络中的所有链路通过在棋盘上寻找骑士巡游路径的方法来获得。如果网络中存在链路故障，那么网络中的链路则不能够在棋盘上寻找到一条完整的骑士巡游路径，因此通过一定大小的棋盘建立网络故障定位模型后，利用故障监测和定位算法能够解决网络中的单链路故障定位问题。




(a) 5×6 棋盘上骑士巡游路径　　(b) 4×6 棋盘上骑士巡游路径

图 3.2 棋盘上的合法骑士巡游路径

## 3.2 单故障定位策略

由于网络拓扑、无向图、棋盘三者之间存在一定的联系，因此根据骑士巡游理论，本章采用棋盘建立网络故障定位模型，然后提出一种网络节点度数感知的分裂方法，将网络中的所有节点、链路一一映射到相应大小的$m\times n$棋盘上，并且能够反映网络中所有节点与链路之间的连接关系，同时构造网络与棋盘的静态映射表；最后利用探测信号返回网络中的故障链路信息，通过查找静态映射表定位出故障链路的准确位置。本节将光网络中基于骑士巡游定位单链路故障的策略分为网络节点度数感知的分裂方法和故障链路定位两个过程。

### 3.2.1 节点度数感知的分裂方法

由于网络拓扑中的节点与链路之间的连接关系可以通过无向图进行描述，对于含有$N$个网元节点和$L$条链路的网络拓扑，若该网络拓扑用无向图$G(V,E)$来表示，其中$V$和$E$分别表示无向图$G$的顶点数和边数，因此$V=N$，$E=L$，通过下面的两个定理可以清楚地表达出无向图$G$中顶点与边的关系。

定理 1 每个图中，节点度数的总和等于边数的二倍，即$\sum_{x\in v}\deg(x)=2|E|$[47]。

证明：设图$G$中有$m$条边，每条边均连接两个节点，即给每个节点的度数各加1，$m$条边总共提供$2m$度数，即$\sum_{x\in v}\deg(x)=2m$。

定理 2 每个图中，度数为奇数的节点个数必定是偶数[47]。

证明：设$V_1$，$V_2$分别是$G$中奇数度数和偶数度数的节点集，则由定理 1 知

$\sum_{x\in v_1}\deg(x)+\sum_{x\in v_2}\deg(x)=\sum_{x\in v}\deg(x)=2|E|$，由于$\sum_{x\in v_2}\deg(x)$为偶之和，必为偶数，而$2|E|$是偶数，故得$\sum_{x\in v_1}\deg(x)$为偶数，即$|V_1|$是偶数。

在将网络中的所有节点和链路一一映射到棋盘上时，首先应该以骑士能够遍历棋盘上所有的方格为前提条件，然后重构网络的逻辑拓扑，将网络中的各个节点和链路分别映射到棋盘上的方格以及两个方格之间的合法马步，且改造后的网络拓扑状态不改变网络的原有属性。首先将网络拓扑中的节点、链路关系用无向图$G(V,E)$进行描述，同时对图中的每个节点进行度数感知分裂，并利用故障定位的网络模型的思想，将分裂后的节点映射到一定大小的$m\times n$棋盘上。因此，根据网络中节点、链路与棋盘的映射关系，提出了节点度数感知分裂的重构网络逻辑拓扑的算法，其过程如下所示：

(1) 分析网络中各个节点$i$的度数$d(i)$，并判断各个节点度数是否为偶数度，若节点度数为偶数则记为$even(i)$，并将偶数个数$evenN$加 1；若节点度数为奇数则记为$odd(i)$，并将奇数个数$oddN$加 1。

(2) 根据定理 2 可知，网络中度数为奇数的节点必定是偶数，则任意选取其中的两个奇数度节点分别作为源节点和目的节点，然后对剩余的奇数度节点进行链路构造，使其节点度数为偶数；或者将所有奇数度节点进行链路构造，使网络中所有节点的度数均为偶数度，并任意选取网络中的两个节点作为源节点和目的节点。

(3) 分析度数为奇数的两个节点之间是否为单跳连通，若这两个奇数度节点单跳连通，则在这两个节点之间添加一条虚拟链路，并记为$addlink(i)$，同时将这两个节点的度数分别加 1；若需要构造的两个奇数度节点非单跳连通，则寻找一条连接这两个节点需要的链路数最少的路径$p(i,j)$，并将这两个节点的度数分别加 1，同时将这条路径经过的节点的度数分别加 2，然后重新统计网络中各个节点的度数$d(i)$，并根据节点度数进行降序排列。

(4) 根据网络的节点连通度对网络节点进行节点度数感知分裂，若节点$i$的度数$d(i)$为偶数，则该节点分裂的个数为$d(i)/2$；若节点$i$的度数$d(i)$为奇数，则该节点分裂的个数为$(d(i)+1)/2$，并统计网络节点分裂后总的节点个数$nodeNum$的值。

(5) 首先对网络中度数$d(i)$最大的节点$i$进行节点分裂，若存在多个节点的度数相同，则先对节点标号最小的节点进行节点分裂，在节点分裂过程中若遇到虚拟链路，则在节点分裂时，首先连接该条虚拟链路。按照这种方法依次对网络中的所有节点进行分裂，直到网络中所有节点分裂完毕。

(6) 检查分裂后的节点链路是否存在多条路径或者环路，若存在环，则对环中分裂后的节点与原节点的链路互换其中的一条连接链路，检查此时的链路连接中是

否还存在环，若不存环链路，则从源节点开始检查是否存在一条经过分裂后的所有节点并且包含所有链路的路径，若存在这样的路径 $path$，则划分出该条路径，并对路径上的所有链路从源节点到目的节点进行编号 $path(i)$。

根据上述的节点度数感知的分裂方法，可以对任何网络进行相应的重构网络逻辑拓扑的分析，并且能够寻找到需要的一条从源节点到目的节点的路径。

### 3.2.2 故障链路定位

对网络进行节点度数感知分裂后，需要将改造后的网络中的所有节点和链路一一映射到一个 $m\times n$ 棋盘上，然后在网络的源节点和目的节点周期性发送一个带有网络节点编号信息的探测信号，实时监测网络中的链路状态。因此，在构造相应的骑士巡游棋盘后，定位网络中单个链路故障的步骤如下所示：

(1) 分析网络节点分裂后的总的节点个数 $nodeNum$，构造一个大小为 $m\times n$ 的棋盘，其中 $m\times n$ 的积大于等于 $nodeNum$，并且是接近于 $nodeNum$ 的偶数。

(2) 在 $m\times n$ 棋盘上寻找一条骑士巡游路径，该条路径为哈密顿路或哈密顿回路均可以。

(3) 将路径 $path$ 中的节点编号 $path(i)$ 映射到寻找到的骑士巡游路径中，若 $nodeNum$ 等于 $m\times n$ 的积，则直接进行节点映射；若 $nodeNum$ 小于 $m\times n$ 的积，则在路径 $path$ 的终节点前增加 $(m\times n)-(nodeNum)$ 个虚拟节点，然后进行节点映射，同时进行节点编号，构造网络的静态映射表。

(4) 在源节点和目的节点同时周期性的发送携带网络节点信息的探测信号，按照骑士巡游行走路线进行网络监测，当发现网络故障时，探测信号则返回节点的编号信息，因此根据源节点的静态映射表可以立即判断网络中的哪条链路发生故障。

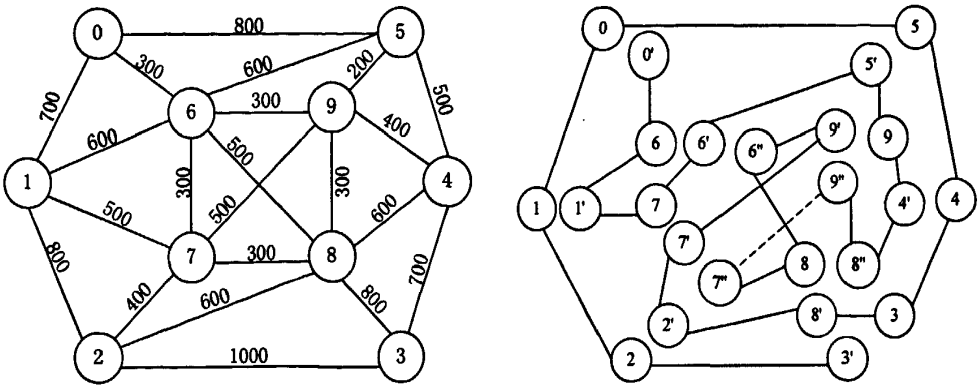
如图 3.3 所示，为利用骑士巡游理论对网络中的单个链路故障进行定位的伪代码，二维数组 $adjnet[N][N]$ 用来存储网络的邻接矩阵，其中 $N$ 表示网络中的节点个数，数组 $degree[]$、$node[][]$ 分别用于存储网络中每个节点的度数和节点度数感知分裂后的节点编号，一维数组 $path[]$ 用于存储网络节点度数感知分裂后寻找到的一条包含所有链路的路径，数组 $kt_path[]$ 用来存储 $m\times n$ 棋盘上骑士巡游路径的序号，$fault_label$ 是探测信号返回的网络故障链路的编号值，$findnet_path()$ 是网络节点度数分裂后寻找一条包含所有链路的路径函数，$look_maptable()$ 函数表示网络中的每条链路与棋盘上骑士巡游路径的映射表。

```
Create the adjacency matrix of the network adjnet[N][N]
Initialize empty arrays degree[],node[][],path[],kt_path[]
for i from 1 to N
    set count<=0
    for j from 1 to N
        set count<=the non-zero number of the adjnet[i][j]
    set degree[i]<=count
for i from 1 to N
    set nodeNum<=(nodeNum+|degree[i]|/2)
    for j from 0 to (|degree[i]|/2-1)
        set node[i][j]<=j
findnet_path(path[i])
if nodeNum<=(m×n) then
    construct the chessboard m×n
for i from 1 to m×n
    label the knight's tour path serial kt_path[i]
for i from 1 to m×n
        set kt_path[i]<= path[i]
for i from 1 to m
    for j from 1 to n
        set fault_label <=i×n+j+1
look_maptable(fault_label)
return 1
```

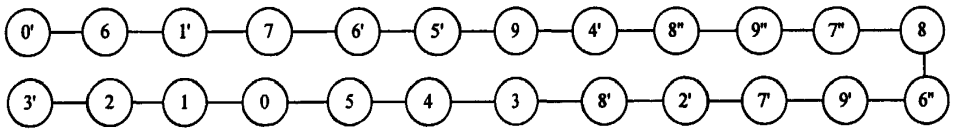
图 3.3 故障定位算法的伪代码

## 3.3 仿真分析

为了验证骑士巡游理论是否能够对网络中的单个链路故障进行准确、快速的定位，基于骑士巡游理论，本章提出了一种网络节点度数感知的分裂方法和故障链路定位方法判定网络中出现的单链路故障情况，根据上述的网络节点度数感知的分裂方法，本节对 mesh 网络中的 Small、NSF、ARPA2、BELLCORE 四种典型网络以及 COST239 网络进行了验证。图 3.4(a)为 Small 网络拓扑，图 3.4(b)是 Small 网络进行节点度数感知分裂后的网络结构，网络的节点个数从10个节点增加到24个节点，同时在网络中的节点 7 和 9 之间添加了一条虚拟链路（虚线表示的链路），使网络中原有的 22 条链路增加到 23 条，其中虚拟链路可以看成是原来的两个节点之间已经存在的链路；图 3.4(c)是 Small 网络根据节点度数感知分裂后寻找到的一条遍历网络中所有节点和链路的路径。



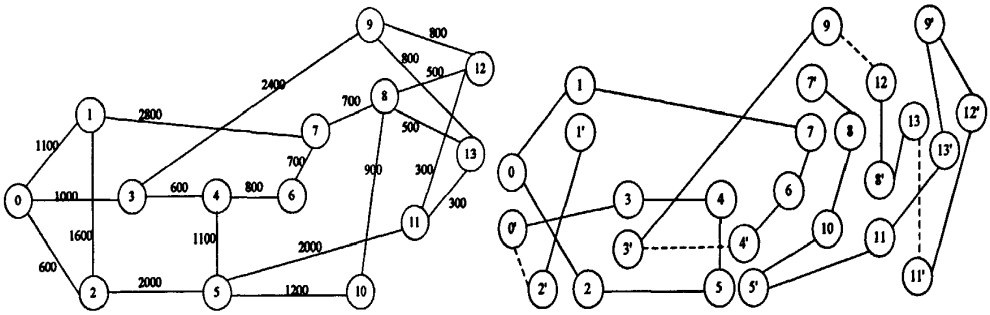
(a)网络拓扑：10 个节点，22 条链路　(b)节点度数感知分裂后：24 个节点，23 条链路



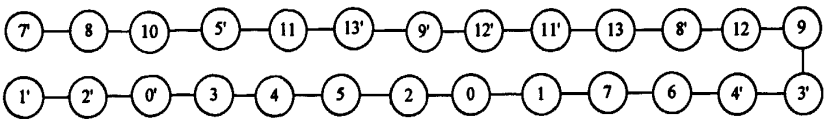
(c)网络节点度数感知分裂后的一条 *path* 路径

图 3.4 Small 网络拓扑及其节点度数感知分裂

图 3.5 到图 3.8 分别对 NSF、COST239、ARPA2 和 BELLCORE 四种网络进行了节点度数感知分裂，并且给出了这四种网络的拓扑结构及其节点度数感知分裂后的网络拓扑结构以及寻找到的一条 *path* 路径。仿真结果如图 3.5 到图 3.8 所示，其中每个网络分裂后的节点个数都增加了许多，并且根据网络中奇数度节点的个数及其网络结构增加了一定数量的虚拟链路。

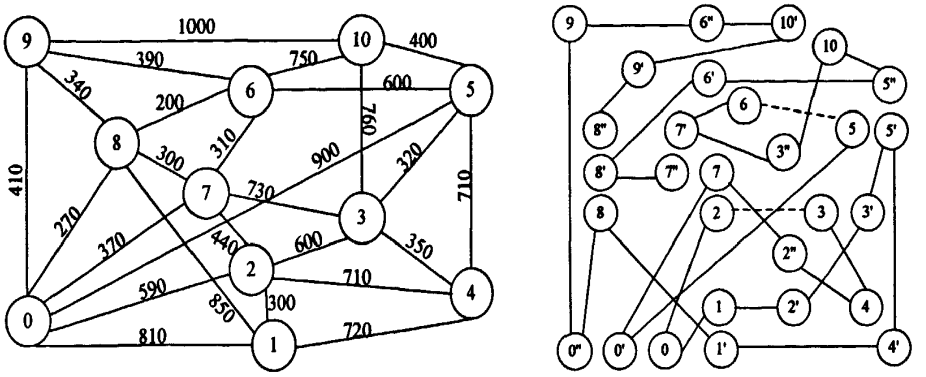


(a)网络拓扑：14 个节点，21 条链路　(b)节点度数感知分裂后：26 节点，25 条链路

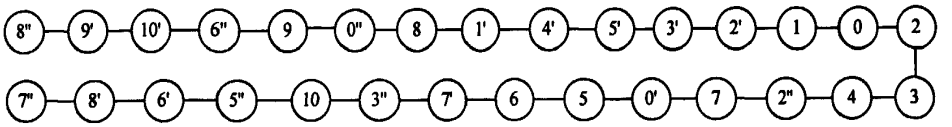


(c)网络节点度数感知分裂后的一条 *path* 路径

图 3.5 NSF 网络拓扑及其节点度数感知分裂

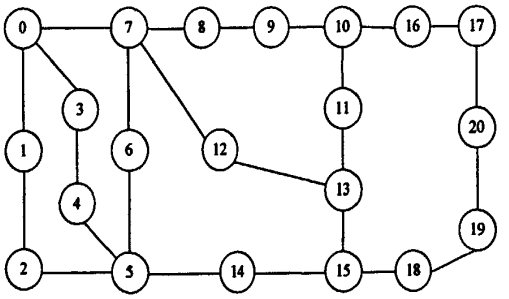



(a)网络拓扑：11 个节点，26 条链路　　(b)节点度数感知分裂后：29 节点，28 条链路

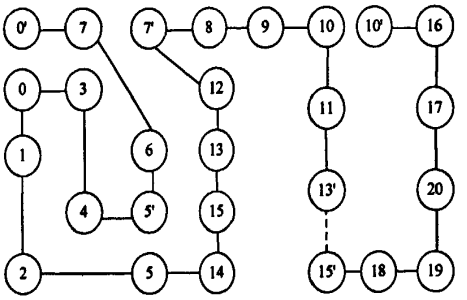



(c)网络节点度数感知分裂后的一条 *path* 路径

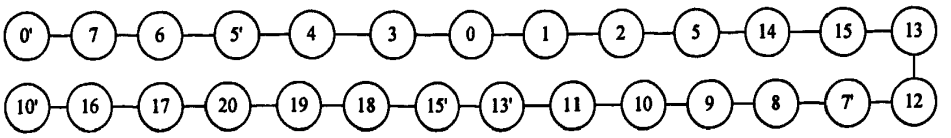
图 3.6 COST239 网络拓扑及其节点度数感知分裂




(a)网络拓扑：21 个节点，25 条链路

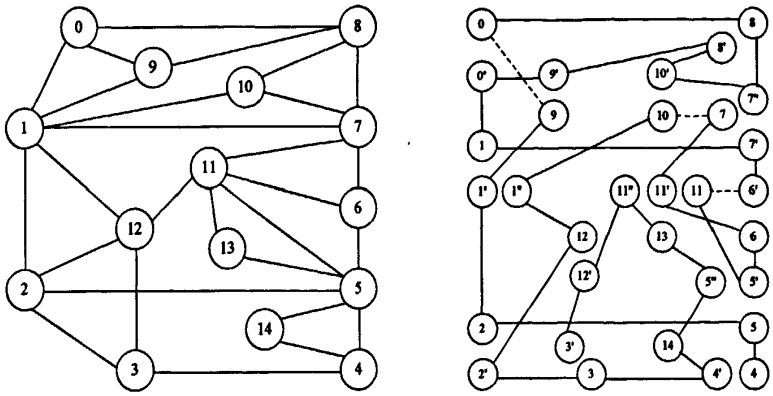



(b)节点度数感知分裂后：27 节点，26 条链路

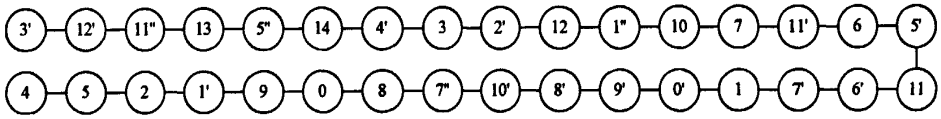



(c)网络节点度数感知分裂后的一条 *path* 路径

图 3.7 ARPA2 网络拓扑及其节点度数感知分裂



(a)网络拓扑：15 个节点，28 条链路　　(b)节点度数感知分裂后：32 节点，31 条链路



(c)网络节点度数感知分裂后的一条 *path* 路径

图 3.8 BELLCORE 网络拓扑及其节点度数感知分裂

表 3.1 清楚地说明了网络拓扑进行节点度数感知分裂前后的网络节点个数及其虚拟链路的变化情况，从表中可知，除 ARPAR2 网络外，网络进行节点度数感知分裂后，每个网络的节点个数均大幅度地增加。针对网络中的虚拟链路的个数，需要根据网络中奇数度节点的个数，以及这些奇数度节点之间是否存在单跳连通有关，若两个奇数度节点之间不存在单跳连通，网络进行节点度数感知分裂过程中则需要增加相对较多的虚拟链路，而在文中提到的五种网络拓扑结构中，奇数度节点之间存在单跳连通，故需要添加的虚拟链路数相对较少。

表 3.1 节点度数感知分裂前后的网络变化

| 网络 / 数量 | Small | NSF | COST239 | ARPA2 | BELLCORE |
|---|---|---|---|---|---|
| 分裂前节点 | 10 | 14 | 11 | 21 | 15 |
| 分裂后节点 | 24 | 26 | 29 | 27 | 32 |
| 虚拟链路 | 1 | 4 | 2 | 1 | 3 |

对网络进行节点度数感知分裂后，寻找一条经过所有节点和链路的路径，然后根据分裂后的网络节点个数构造相应大小的骑士巡游棋盘，并对网络中的所有节点、链路与棋盘进行一一映射，形成对应的静态映射表。从图 3.4 中表明 Small 网络进行节点度数感知分裂后有 24 个节点和 23 条链路，故可将分裂后的 Small 中的所有节点和链路映射到 4×6 棋盘上。表 3.2 为 Small 网络与 4×6 棋盘的静态映射表，从表中可以看出网络中的每条链路以及分裂后的每个节点均有唯一的网络编号，因此当网络中发生链路故障时，根据探测信号返回的故障链路编号信息，通过查找静态

映射表，可以唯一确定故障链路发生的准确位置。

表 3.2 Small 网络与 4×6 棋盘的映射表

| 编号 | 对应节点 | 对应链路 | 编号 | 对应节点 | 对应链路 | 编号 | 对应节点 | 对应链路 |
|---|---|---|---|---|---|---|---|---|
| 1 | 0’ | 0-6 | 9 | 8” | 8-9 | 17 | 8’ | 8-3 |
| 2 | 6 | 6-1 | 10 | 9” | 9-7 | 18 | 3 | 3-4 |
| 3 | 1’ | 1-7 | 11 | 7” | 7-8 | 19 | 4 | 4-5 |
| 4 | 7 | 7-6 | 12 | 8 | 8-6 | 20 | 5 | 5-0 |
| 5 | 6’ | 6-5 | 13 | 6” | 6-9 | 21 | 0 | 0-1 |
| 6 | 5’ | 5-9 | 14 | 9’ | 9-7 | 22 | 1 | 1-2 |
| 7 | 9 | 9-4 | 15 | 7’ | 7-2 | 23 | 2 | 2-3 |
| 8 | 4’ | 4-8 | 16 | 2’ | 2-8 | 24 | 3’ | \ |

表 3.3 为 NSF 网络与棋盘的静态映射表，由于 NSF 网络进行节点度数感知分裂后只有 26 个节点，与其最接近且大于其值的偶数为 28，因此将节点度数感知分裂后的所有节点和链路映射到 4×7 棋盘上，同时在链路 2-1 之间添加两个虚拟节点，用以满足棋盘上的方格总数，在定位网络的单个链路故障时，此种方法对故障定位的准确性不产生任何影响。从表中可以看出，除 2-1 链路外，其它每条链路均有唯一的网络编号，因此可以根据探测信号返回的故障链路编号信息，可以唯一确定发生故障的链路，而当探测信号返回的编号信息为 25、26 或 27 时，表明链路 2-1 之间发生故障。

表 3.3 NSF 网络与 4×7 棋盘的映射

| 编号 | 对应节点 | 对应链路 | 编号 | 对应节点 | 对应链路 | 编号 | 对应节点 | 对应链路 |
|---|---|---|---|---|---|---|---|---|
| 1 | 7’ | 7-8 | 11 | 8’ | 8-12 | 21 | 5 | 5-4 |
| 2 | 8 | 8-10 | 12 | 12 | 12-9 | 22 | 4 | 4-3 |
| 3 | 10 | 10-5 | 13 | 9 | 9-3 | 23 | 3 | 3-0 |
| 4 | 5’ | 5-11 | 14 | 3’ | 3-4 | 24 | 0’ | 0-2 |
| 5 | 11 | 11-13 | 15 | 4’ | 4-6 | 25 | 2’ | 2-2(1) |
| 6 | 13’ | 13-9 | 16 | 6 | 6-7 | 26 | 2’(1) | 2(1)-2(2) |
| 7 | 9’ | 9-12 | 17 | 7 | 7-1 | 27 | 2’(2) | 2(2)-1 |
| 8 | 12’ | 12-11 | 18 | 1 | 1-0 | 28 | 1 | \ |
| 9 | 11’ | 11-13 | 19 | 0 | 0-2 | | | |
| 10 | 13 | 13-8 | 20 | 2 | 2-5 | | | |

表 3.4 COST239 网络与 5×6 棋盘的映射表

| 编号 | 对应节点 | 对应链路 | 编号 | 对应节点 | 对应链路 | 编号 | 对应节点 | 对应链路 |
|---|---|---|---|---|---|---|---|---|
| 1 | 8” | 8-9 | 11 | 3’ | 3-2 | 21 | 5 | 5-6 |
| 2 | 9’ | 9-10 | 12 | 2’ | 2-1 | 22 | 6 | 6-7 |
| 3 | 10’ | 10-6 | 13 | 1 | 1-0 | 23 | 7’ | 7-3 |
| 4 | 6” | 6-9 | 14 | 0 | 0-2 | 24 | 3” | 3-10 |
| 5 | 9 | 9-0 | 15 | 2 | 2-3 | 25 | 10 | 10-5 |
| 6 | 0” | 0-8 | 16 | 3 | 3-4 | 26 | 5” | 5-6 |
| 7 | 8 | 8-1 | 17 | 4 | 4-2 | 27 | 6’ | 6-8 |
| 8 | 1’ | 1-4 | 18 | 2” | 2-7 | 28 | 8’ | 8-7(1) |
| 9 | 4’ | 4-5 | 19 | 7 | 7-0 | 29 | 7”(1) | 7(1)-7 |
| 10 | 5’ | 5-3 | 20 | 0’ | 0-5 | 30 | 7” | \ |

表 3.5 ARPA2 网络与 4×7 棋盘的映射表

| 编号 | 对应节点 | 对应链路 | 编号 | 对应节点 | 对应链路 | 编号 | 对应节点 | 对应链路 |
|---|---|---|---|---|---|---|---|---|
| 1 | 0’ | 0-7 | 11 | 14 | 14-15 | 21 | 15’ | 15-18 |
| 2 | 7 | 7-6 | 12 | 15 | 15-13 | 22 | 18 | 18-19 |
| 3 | 6 | 6-5 | 13 | 13 | 13-12 | 23 | 19 | 19-20 |
| 4 | 5’ | 5-4 | 14 | 12 | 12-7 | 24 | 20 | 20-17 |
| 5 | 4 | 4-3 | 15 | 7’ | 7-8 | 25 | 17 | 17-16 |
| 6 | 3 | 3-0 | 16 | 8 | 8-9 | 26 | 16 | 16-10(1) |
| 7 | 0 | 0-1 | 17 | 9 | 9-10 | 27 | 10’(1) | 10(1)-10 |
| 8 | 1 | 1-2 | 18 | 10 | 10-11 | 28 | 10’ | \ |
| 9 | 2 | 2-5 | 19 | 11 | 11-13 | | | |
| 10 | 5 | 5-14 | 20 | 13’ | 13-15 | | | |

表 3.4 到表 3.6 分别为 COST239、ARPA2、BELLCORE 网络与棋盘的静态映射表，其中 COST239、ARPA2 网络在进行节点度数感知分裂后分别有 29、27 个节点，因此为这两个网络分别增加一个虚拟节点后，可将其分别映射到 5×6、4×7 棋盘上。而进行节点度数感知分裂后的 BELLCORE 网络有 32 个节点，因此不需要增加虚拟节点就可将其直接映射到 4×8 棋盘上。

表 3.6 BELLCORE 网络与 4×8 棋盘的映射表

| 编号 | 对应节点 | 对应链路 | 编号 | 对应节点 | 对应链路 | 编号 | 对应节点 | 对应链路 |
|---|---|---|---|---|---|---|---|---|
| 1 | 3’ | 3-12 | 12 | 10 | 10-7 | 23 | 8’ | 8-10 |
| 2 | 12’ | 12-11 | 13 | 7 | 7-11 | 24 | 10’ | 10-7 |
| 3 | 11” | 11-13 | 14 | 11’ | 11-6 | 25 | 7” | 7-8 |
| 4 | 13 | 13-5 | 15 | 6 | 6-5 | 26 | 8 | 8-0 |
| 5 | 5” | 5-14 | 16 | 5’ | 5-11 | 27 | 0 | 0-9 |
| 6 | 14 | 14-4 | 17 | 11 | 11-6 | 28 | 9 | 9-1 |
| 7 | 4’ | 4-3 | 18 | 6’ | 6-7 | 29 | 1’ | 1-2 |
| 8 | 3 | 3-2 | 19 | 7’ | 7-1 | 30 | 2 | 2-5 |
| 9 | 2’ | 2-12 | 20 | 1 | 1-0 | 31 | 5 | 5-4 |
| 10 | 12 | 12-1 | 21 | 0’ | 0-9 | 32 | 4 | \ |
| 11 | 1” | 1-10 | 22 | 9’ | 9-8 | | | |

表 3.7 网络中故障链路的定位时间(ms)

| ID | OSPF | Small | NSF | COST239 | ID | OSPF | Small | NSF | COST239 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 40 | 1.042 | 3.271 | 1.589 | 16 | 40 | 27.57 | 54.673 | 31.168 |
| 2 | 40 | 4.206 | 7.477 | 6.262 | 17 | 40 | 24.766 | 51.402 | 29.533 |
| 3 | 40 | 6.542 | 13.084 | 9.766 | 18 | 40 | 21.028 | 38.318 | 26.215 |
| 4 | 40 | 7.944 | 22.43 | 11.589 | 19 | 40 | 17.757 | 33.374 | 24.159 |
| 5 | 40 | 10.748 | 23.832 | 13.505 | 20 | 40 | 15.421 | 30.374 | 22.430 |
| 6 | 40 | 11.682 | 27.57 | 14.766 | 21 | 40 | 11.682 | 21.028 | 18.224 |
| 7 | 40 | 13.551 | 31.308 | 18.738 | 22 | 40 | 8.411 | 15.888 | 15.421 |
| 8 | 40 | 16.355 | 32.71 | 22.103 | 23 | 40 | 4.673 | 13.084 | 13.972 |
| 9 | 40 | 17.757 | 34.112 | 25.421 | 24 | 40 | \ | 8.411 | 10.561 |
| 10 | 40 | 20.093 | 36.449 | 26.916 | 25 | 40 | | 7.477 | 7.009 |
| 11 | 40 | 21.495 | 38.785 | 29.720 | 26 | 40 | | 5.14 | 5.140 |
| 12 | 40 | 23.832 | 42.523 | 31.121 | 27 | 40 | | 2.084 | 2.336 |
| 13 | 40 | 25.234 | 53.738 | 34.907 | 28 | 40 | | \ | 1.402 |
| 14 | 40 | 27.57 | 56.542 | 36.729 | 29 | 40 | | | 0.701 |
| 15 | 40 | 29.434 | 58.411 | 33.972 | 30 | 40 | | | \ |

表 3.7 为利用骑士巡游理论对 Small、NSF、COST239 网络中各条可能出现的故障链路的定位时间，由于在链路故障定位的过程中，并未考虑对网络中链路权重的

大小进行顺序遍历，因此并未对最大的定位时间进行限制。其中，除了 NSF 网络的最大定位时间超过 OSPF[39]协议的故障定位时间 40ms 外，Small、COST239 网络中故障的最大定位时间均小于 40ms，而且这三种网络的平均定位时间分别为 16.05ms、30.243ms、18.763ms 均低于 OSPF 协议的 40ms，因此只需要将网络的链路权重等约束性条件考虑到节点度数感知分裂的过程中，则可能降低最大的故障链路定位时间。

文献[20]中指出圈覆盖算法定位故障链路的代价为监测圈数目和链路覆盖长度的成本之和，假设一个网络拓扑用无向图 $G(V,E)$ 来表示，其中 $V$ 是网络节点，$E$ 表示网络中的链路。因此根据圈覆盖算法，网络可以被分解成一组圈的集合，假设用 $C=\{c_1,c_2,...,c_M\}$ 表示图 $G(V,E)$ 被分解后的圈的集合，其中 $c_i$ 表示圈集合中的第 $i$ 个监测圈，若图 $G$ 中的每条边 $e\in E$，则用 $C(e)=|\{i:e\in c_i\}|$ 表示圈 $C$ 中包含链路 $e$ 的个数，因此网络的链路覆盖长度为：

$$len(C)=\sum_{i=1}^{M}|c_i|=\sum_{j=1}^{L}C(e_j) \quad 式(3.1)$$

而本章的单链路故障定位策略，只需要利用探测信号实时监测网络中的链路状态并返回相关信息，通过查找静态映射表对故障链路进行定位，假设用 $L$ 表示网络中的链路数，$n$ 为节点分裂后增加的虚拟链路总数，因此链路覆盖长度为：

$$len(l)=L+\sum_{i=1}^{n}addlink(i) \quad 式(3.2)$$

表 3.8 为使用圈覆盖算法和本文算法对 Small、NSF 等五种网络的链路覆盖长度的比较，从表中可以看出，圈覆盖算法比本文算法的链路覆盖长度相对较多，因此本文的故障定位策略相对地降低了网络资源的开销。

表 3.8 链路覆盖长度

| | Small | NSF | COST239 | ARPA2 | BELLCORE |
|---|---|---|---|---|---|
| 圈覆盖算法 | 43 | 40 | 56 | 40 | 55 |
| 本文算法 | 23 | 25 | 28 | 26 | 31 |

## 3.4 本章小结

本章将骑士巡游理论引入到光网络故障管理中，并提出一种解决网络中单链路故障定位问题的策略。该策略首先根据骑士巡游理论，建立网络的故障定位模型，并且提出一种节点度数感知的分裂方法重构网络的逻辑拓扑，同时构造网络与棋盘的静态映射表；然后利用探测信号返回网络中的故障链路信息，通过查找静态映射表判断该故障链路的准确位置。同时对 mesh 网络中的经典网络拓扑进行了分析，

例如对 Small、NSF、COST239、ARPA2、BELLCORE 网络中可能出现的单个链路故障情况进行分析和定位，并从链路故障定位时间和链路覆盖长度两个方面进行了分析。仿真结果表明，所提出方法的平均链路故障定位时间小于 OSPF 协议的故障定位时间 40ms，链路覆盖长度相对小于圈覆盖算法的链路覆盖长度。由于本章提出的定位策略是验证骑士巡游理论是否能够应用到光网络中的故障链路定位，因此在仿真时并未考虑链路权重对故障定位时间的影响，然而可以通过增加一些约束性条件，在保证故障定位率的同时，可以进一步减小网络故障定位的最大时间。

# 第四章 基于骑士巡游的多链路故障定位机制

人们对网络业务的需求远远超过了各项保障网络生存性技术的更新速度，由于光网络数据传输的高速性，使得网络发生故障时将产生巨大的经济损失和社会影响，因此对于保障网络传输的稳定性和安全性提出了更高的要求。而故障监测和定位是保障网络生存性的重要前提条件，因此准确、快速的故障定位技术是降低因网络故障所引起损失的重要保障，才能提高整个网络的有效性和可靠性。

针对光网络中的单故障定位技术，现有的故障管理文献中均对网络中的故障进行了分析，并根据相应的算法或机制进行故障探测和定位，同时也阐述了故障定位机制在网络管理中的重要作用，即故障恢复和保护的前提，并提出为了更好的利用网络资源，必须找出关于定位网络故障的有效算法或机制，其中较多的文献是对现有故障监测和定位方法进行的性能优化，从而达到降低网络监测成本的目的。然而现有关于故障管理的文献中对网络的多故障问题没有较多的研究，当网络发生多故障时，这些单故障定位方法针对网络多故障并不能给出快速、有效地判断，不能给予网络的可靠性提供有效的保障。因此，鉴于网络中多故障定位技术的重要性，本章提出了一种准确、迅速的多故障定位机制。

根据骑士巡游理论，本章提出一种多链路故障定位机制，该定位机制首先建立多故障链路定位的网络模型，并将网络中的所有节点映射到一个带有漏洞的$m\times n$棋盘上；然后在备选探测信号节点周期性发送探测信号信息，目的节点将收到的有效光路存储在链表中；最后通过链路匹配计算网络中出现的故障链路。仿真结果表明该定位机制能够有效地定位网络中出现的多条链路故障，同时计算了该定位机制的时间和空间复杂度。

## 4.1 多故障定位的网络模型

图是一个数学抽象模型，通过图可以表征客观世界中某些具体事物之间的联系，因此可以使用一个二元关系的关系图来表示网络中的节点和链路之间的连接关系，而不需要考虑节点的位置和链路的长度。对于一个包含$V$个节点和$E$条链路的网络来说，可以使用二元关系的无向图$G(V,E)$进行描述，通过构造邻接矩阵来表示该网络中各个节点与各条链路之间的连接关系。根据图的特性，棋盘上的方格和方格之间的连接也可以分别用点和连线来表示，因此，利用无向图也能够表征棋盘上的方格和方格之间的连接关系，其中，棋盘上的方格以及方格之间的连接关系可以分

别用无向图的顶点和边来表示。由此可见，通过一定的方法，可以将网络拓扑中的节点和链路之间的连接关系反映到棋盘上，并且不改变网络的原有属性。根据骑士巡游理论，本章将网络中的所有节点和链路分别用棋盘的方格以及方格之间的合法马步来表示，通过骑士巡游行走路线来监测网络中的链路状态，从而解决网络中的多链路故障定位问题。

骑士巡游理论[42]为马遍历棋盘问题，骑士（马）的遍历是为马能走遍棋盘的每一个方格且每格恰好经过一次，还可以进一步要求马遍历整个棋盘后恰能回到出发点，这样的一条路径称为马的回路。而马步图则是将棋盘中的方格作为顶点的集合，骑士的合法马步作为边的集合构成的一个简单无向图，因此通常将马的遍历问题转化成图论问题进行分析，可以把棋盘的格子作为图的顶点，相互之间可以经马步到达的格子之间以边相连[42]-[44]。

根据 3.1 故障定位的网络模型中对骑士巡游理论的介绍，可知骑士遍历 $m\times n$ 棋盘时，若骑士位于第 $i$ 行 $j$ 列格记为 $a(i,j)$，并令 $(i,j)$ 为其当前位置，由于马的合法棋步是 $(i\pm 2, j\pm 1)$ 或 $(i\pm 1, j\pm 2)$，因此当 $a_0 = i+j$ 时，马的下一个合法位置是 $a_1 = i+j\pm 3$ 或 $a_1 = i+j\pm 1$。所谓有洞的棋盘[46][48]是指将棋盘上的某些方格设置为骑士禁止访问的位置，这些被禁止访问的方格则被称为棋盘的漏洞。因此将网络的节点和链路通过带有漏洞的棋盘进行描述时，具有更好的灵活性，可以将网络与棋盘映射时不需要的或者多余的方格设置为禁止访问的位置，从而降低骑士遍历棋盘的复杂度。如图 4.1 所示为在带有中心漏洞的 3×3 棋盘上遍历所有方格的骑士巡游路线图，其中灰色的方格为骑士禁止访问的位置，该图说明将棋盘上的任意方格作为骑士巡游的起始位置都能够找到一条哈密顿回路，若将棋盘中其它位置的方格置为禁止访问方格，则不能够寻找到一条遍历棋盘上所有方格的骑士巡游路径，从而在网络拓扑与棋盘关系的映射时，可以根据需求将棋盘上的方格设置为禁止访问位置，从而增加两者之间映射关系的灵活性。

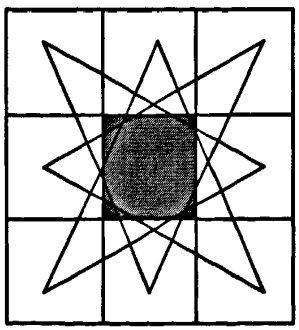


图 4.1 带有中心漏洞的 3×3 棋盘上的骑士巡游路线

如图 4.2 所示，为在带有漏洞的 8×8 棋盘上寻找到的骑士巡游路径的马步顺序图，方格中的数字为骑士遍历棋盘时的马步顺序数值，标示 1 的方格为骑士巡游的

起始位置，图中灰色的方格为骑士禁止访问的位置，即棋盘的漏洞。如图 4.2 所示，在带有漏洞的 8×8 棋盘上，图 4.2(a)为一条骑士巡游回路，而图 4.2(b)是一条骑士巡游开路的路径，并且两者之间在棋盘上禁止访问的方格位置也不尽相同，因此可以根据不同的需求在棋盘上设置禁止访问的位置来达到网络与棋盘之间所需要的映射关系，从而寻找需要的骑士巡游路径。

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 16 | 23 | 26 | 39 | | | | |
| 25 | 38 | 15 | 30 | 27 | 40 | 5 | |
| 22 | 17 | 24 | | 4 | 31 | 28 | |
| 37 | 14 | | | 29 | 6 | 41 | |
| | 21 | 18 | 3 | | | 32 | 43 |
| | 36 | 13 | 20 | | 42 | 7 | 10 |
| | 19 | 2 | 35 | 12 | 9 | 44 | 33 |
| | | | | 1 | 34 | 11 | 8 |

(a)带有漏洞的骑士巡游回路

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | 5 | 26 | 11 | | | |
| | | 12 | | 6 | 25 | | |
| | 16 | 27 | 4 | 13 | 10 | 7 | 24 |
| 28 | 3 | 14 | | | 33 | | 9 |
| 15 | | 17 | | | 8 | 23 | 32 |
| 18 | 29 | 2 | 21 | 34 | 31 | 36 | |
| | | 19 | 30 | | 22 | | |
| | | | 1 | 20 | 35 | | |

(b)带有漏洞的骑士巡游路径

图 4.2 带有漏洞的 8×8 棋盘上的马步顺序

根据网络属性与棋盘之间的联系，通过一定的理论方法可以将网络中的所有节点和链路之间的信息通过棋盘进行描述，本章利用骑士巡游理论可将网络属性反映到棋盘上，其中，网络中任意两个节点之间可能存在的所有路径都能够通过在棋盘上寻找骑士巡游路径的方式来获得。若网络中存在故障链路，那么在棋盘上寻找到的所有的骑士巡游路径中都不存在该条链路映射到棋盘上的马步，因此利用骑士巡游理论构造网络的故障定位模型，能够解决网络中的多链路故障定位问题。

## 4.2 多故障定位机制

在已有的光网络故障监测和定位技术中，比较经典的算法有基于监测设备的圈覆盖算法、监测迹算法等故障定位算法，而这些故障监测和定位算法只是针对光网络中的单链路故障进行分析和定位，而对网络的多故障问题没有比较详细的介绍，并且现有文献中对网络多故障监测和定位的研究也相对较少，根据骑士巡游理论的思想，本章提出了一种适用于光网络的快速多链路故障定位机制。该多链路故障定位机制首先利用网络连通性的特征，使用网元抽象方法将网络中的所有节点映射到带有漏洞的棋盘上，然后利用探测信号收集网络中各条链路的工作状态，最后通过

链路匹配确定网络中发生故障的链路。为了便于分析，将本章所提出的多链路故障机制分为了四个阶段：初始化、网元抽象阶段、探测阶段、匹配阶段。

### 4.2.1 带有约束条件的网元抽象方法

通过图可以表征客观事物之间联系的关系，因此可以使用二元关系图来表示网络、棋盘之间可能存在的某种联系，带有约束条件的网元抽象方法以网络中各个节点的连通度与棋盘上各个方格合法马步数量的关系为依据，合理地将网络中的节点映射到棋盘上的相应位置，网络节点之间的链路用棋盘上两个方格之间的合法马步来表示。将网络中的所有节点映射到棋盘上后，棋盘上多余的方格则被设置为禁止访问方格，并且映射后，棋盘上方格之间的合法马步能够完全反映网络中所有节点、链路之间的连接关系。带有约束条件的网元抽象方法的过程如下所示，其中该过程中包含了网络多故障定位机制的初始化和网元抽象阶段。

(1) 初始化目标网络

1) 通过给出的网络邻接矩阵，统计网络中的各个节点$i$的度数$d(i)$，并根据节点度数的大小按升序排列，同时计算网络中的节点度数总和。

2) 构造一个大小为$m\times n$的棋盘，其中$m\times n$的值大于或等于网络的节点度数总和，同时统计$m\times n$棋盘中各个方格的合法马步个数。

(2) 网元抽象阶段

1) 首先将网络中节点度数最小的节点放到网元抽象集合$P$中，若存在多个度数相同的节点，则先将节点标号最小的节点放入集合$P$中，同时将与$P$中节点相连的节点放入等待集合$S$中。

2) 寻找集合$P$中节点度数$d(i)$与方格的合法马步个数相同或最接近的方格，然后将节点$i$放到该方格中，并将与节点$i$相连的节点分别放到下一跳马步对应的各个方格中，其中节点的度数与其放入的方格的合法马步个数相同或最接近，同时将下一跳马步中多余的方格设置为禁止访问的方格。

3) 寻找与集合$S$中节点相连的网络节点，除$P$中的节点外，将这些节点放到$S$中，同时寻找$S$中节点度数最少并且节点编号最小的节点放到$P$中，根据上一个步骤对$P$中新添加的节点进行网元抽象。

4) 根据上面的网元抽象方法，依次将网络节点进行网元抽象直到集合$P$中包含了网络中的所有节点，然后判断网元抽象后各个节点在棋盘上的合法马步个数是否低于2，若这些节点的合法马步个数低于2，并且节点的合法马步范围内均存在非禁止访问方格，则为其添加一条合法马步。

根据上述步骤中的网元抽象方法，可以对任意网络进行相应的分析，并且能够

建立到一个有效的、带有漏洞的$m \times n$棋盘，进而可以将链路故障定位问题转换成在带有漏洞的棋盘上寻找骑士巡游路径的问题。

### 4.2.2 多链路故障定位

根据带有约束条件的网元抽象方法将网络节点映射到$m \times n$棋盘后，通过周期性发送探测信号，来实时监测网络中所有链路的工作状态，在一定的时间周期内，目的节点对收到的所有光路进行链路匹配，从而判断网络中可能出现的故障链路。其中多链路故障定位过程包含探测阶段、链路匹配阶段，其定位过程如下所示：

(1) 探测阶段

1) 将网络中的所有链路存储在一个网络链表队列 NLQ(Network Links Queue, NLQ)中。
2) 选定网络中的某个节点作为目的节点，同时将其余节点作为备选探测信号的发送节点。
3) 随机选取备选探测信号中的某些节点向目的节点发送探测信号，其中探测信号封装了网络的节点号和路径长度信息。

(2) 链路匹配阶段

1) 目的节点接收从其它节点发送的探测信息，根据光路长度的大小存储在一个链表中，若光路中存在相同长度的路径，则将源节点标号较小的光路放在链表的第一行，并将第一条光路的所有链路存储在一个执行链路队列 ELQ(Executive Links Queue, ELQ)中。其中，存储接收光路的链表中只接收光路长度小于某一定值的链路，若光路长度大于该值，则直接丢弃该条光路。
2) 将链表中其它的光路分别与第一条光路进行链路匹配，如果发现两条光路存在非共享链路，则将非共享链路存储在 ELQ 中。
3) 将 ELQ 中的所有链路与 NLQ 中的链路作比较，若 ELQ 与 NLQ 中的所有链路相同，则说明网络无故障；若 ELQ 的链路长度小于 NLQ 的长度，那么 ELQ 比 NLQ 中缺少的链路则为网络中的故障链路。

图 4.3 为基于骑士巡游理论定位网络中多链路故障的流程图，该流程图包含了在网络中定位多链路故障的初始化、网元抽象、探测和匹配四个阶段。在定位网络故障时，首先需要获取目标网络的网络属性信息，即网络中节点与链路之间的连接关系，利用网元抽象方法对目标网络进行棋盘映射，然后利用探测信号定位网络中出现的故障链路。其中，流程图中的$N_i$表示网络中第$i$个节点的连通度，$H_j$表示棋

盘上第 $j$ 个方格的合法马步个数。

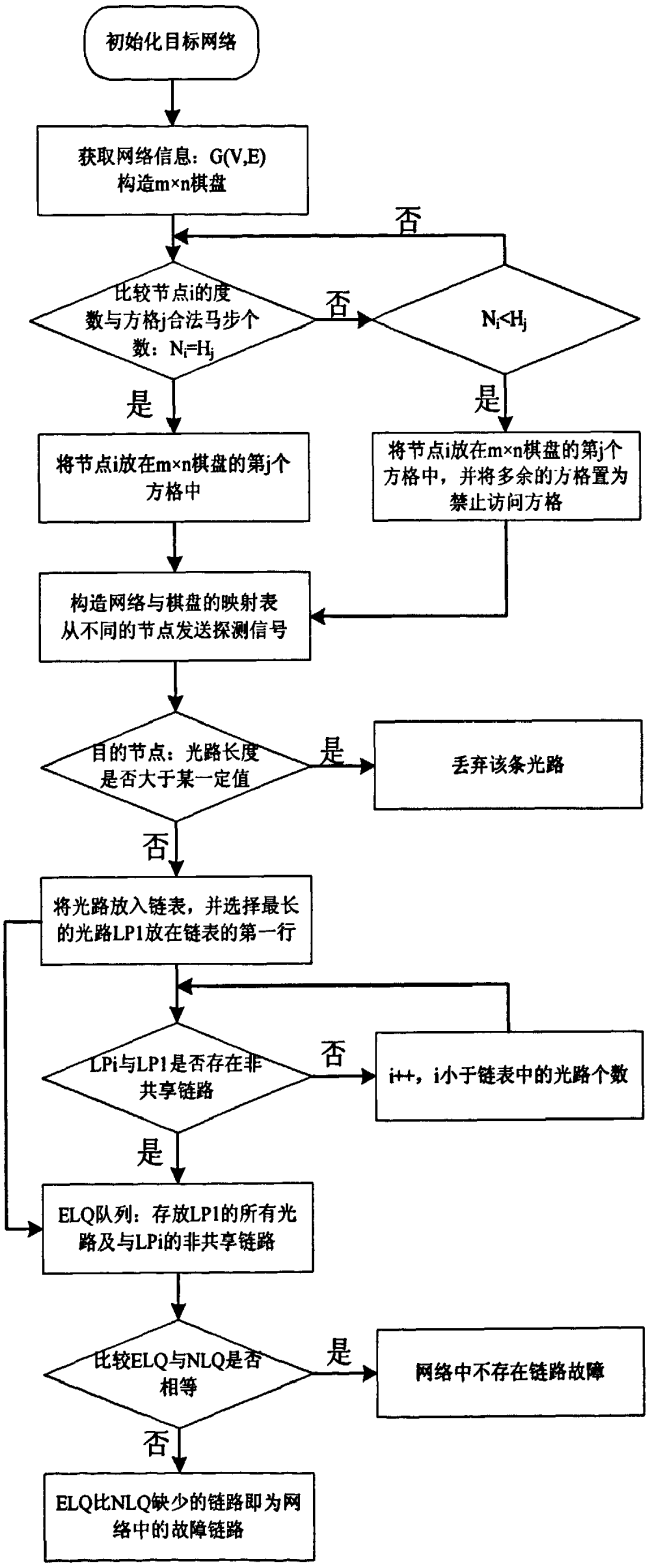



图 4.3 多故障定位流程图

### 4.2.3 算法复杂度分析

对于给定的网络可以使用无向图$G(V,E)$进行描述，其中$V$表示网络的节点数，$E$是网络的链路数。在初始化阶段，为了获取网络信息，包括网络的节点数和链路数，以及网络节点度数总和，并且需要根据网络的大小判断$m\times n$棋盘的大小。则该过程的时间复杂度为$O(V^2)$，$\alpha$为计算时间周期，则该阶段需要的时间复杂度为：

$$T_1=V^2\times\alpha \tag{式(4.1)}$$

在网元抽象过程中，需要计算每个节点与要放入的方格的合法马步个数作比较，在确定某个节点要放入的方格后，还需要比较其邻居节点与方格下一跳方格的合法马步个数。假设$N_i$为网络节点$i$的连通度数，$E[N_i]$为网络的平均节点度数，则该阶段的时间复杂度为：

$$T_2=V\times E[N_i]\times\alpha \tag{式(4.2)}$$

在探测阶段，为了保证目的节点能够收到足够多有效的探测信号，该定位机制采用在网络平均链路长度外增加一跳的方式来保证光路的传播时间。假设$\beta$为邻居节点间的平均传播延迟，$E[d_{sd}]$为网络的平均路径长度，则该阶段的时间复杂度为：

$$T_3=(E[d_{sd}]+1)\times\beta \tag{式(4.3)}$$

在匹配阶段，目的节点需要将链表中的光路逐个进行链路匹配，直到链表中的所有光路都不存在共享链路。假设$n_k$为链表中存储的光路数，若该链表中存储的两条光路长度分别为$L_1$、$L_2$，则计算这两条光路匹配的复杂度为$O(L_1L_2)$，因此链表中两条光路匹配的平均计算复杂度为$O(E[d_{sd}]^2)$，那么该阶段的时间复杂度为：

$$T_4=n_k\times(E[d_{sd}]^2\times\alpha) \tag{式(4.4)}$$

因此，定位网络多链路故障的总的时间复杂度为：

$$\begin{aligned}T&=\sum_{i=1}^{4}T_i\\&=V^2\times\alpha+V\times E[N_i]\times\alpha+(E[d_{sd}]+1)\times\beta+n_k\times(E[d_{sd}]^2\times\alpha)\\&=(V^2+V\times E[N_i]+n_k\times E[d_{sd}]^2)\times\alpha+(E[d_{sd}]+1)\times\beta\end{aligned} \tag{式(4.5)}$$

在定位网络的多链路故障的过程中，需要为网络信息、链表、队列分配一定的存储空间，在初始化阶段的空间复杂度为：

$$S_1=V+m\times n+E \tag{式(4.6)}$$

存储网络中各个顶点度数大小的空间复杂度为$O(V)$，$O(m\times n)$是棋盘中所有方格合法马步个数的空间复杂度；$O(E)$用来存储网络中的所有无向链路，即网络链路队列NLQ的大小。

在匹配阶段，链表中存在$n_k$条光路，每条光路的平均长度为$E[d_{sd}]$，执行链路队列ELQ的空间复杂度为$O(E)$，则该阶段需要的空间复杂度为：

$$S_2 = n_k \times E[d_{sd}] + E \quad \text{式(4.7)}$$

因此，在网络故障定位的过程中需要的总的空间复杂度为：

$$\begin{aligned} S &= S_1 + S_2 \\ &= (V + m \times n + E) + (n_k \times E[d_{sd}] + E) \\ &= V + m \times n + 2E + n_k \times E[d_{sd}] \end{aligned} \quad \text{式(4.8)}$$

## 4.3 性能分析

### 4.3.1 mesh 格状网的算法复杂度

如图 4.4 所示，mesh 格状网中有 M×N 个节点，其中 M、N 分别表示 mesh 格状网的行号和列号，下面对 mesh 格状网的时间和空间复杂度进行了计算和比较。

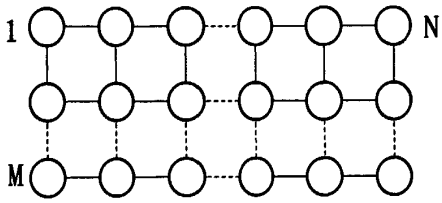



图 4.4 M×N mesh 格状网

假设 mesh 网络中的节点到目的节点对的坐标分别为 $(x_s, y_s)$ 和 $(x_d, y_d)$，源节点到目的节点距离的计算公式为：

$$d_{sd} = |x_d - x_s| + |y_d - y_s| \quad \text{式(4.9)}$$

由于，源节点和目的节点的坐标取值相互独立，因此其坐标的取值概率为：

$$\begin{aligned} P(x_s = m) = P(x_d = n) = \frac{1}{M}, \ \forall m,n = 0,1...M-1 \\ P(y_s = p) = P(y_d = q) = \frac{1}{N}, \ \forall m,n = 0,1...N-1 \end{aligned} \quad \text{式(4.10)}$$

源节点到目的节点距离的取值概率为：

$$\begin{aligned} P(|x_d - x_s| = i) = \sum_k P(x_d = k) P(x_s = k - i) \\ P(|y_d - y_s| = j) = \sum_k P(y_d = k) P(y_s = k - j) \end{aligned} \quad \text{式(4.11)}$$

因此，X 轴和 Y 轴的坐标差值的均值为：

$$\begin{aligned} E[|x_d - x_s|] = \sum_{i=0}^{M-1} i\left(\frac{2(M-i)}{M(M-1)}\right) = \frac{M+1}{3} \\ E[|y_d - y_s|] = \sum_{j=0}^{N-1} j\left(\frac{2(N-j)}{N(N-1)}\right) = \frac{N+1}{3} \end{aligned} \quad \text{式(4.12)}$$

因此，网络中两个节点之间的平均距离为：

$$E[d_{sd}]=E[|y_d-y_s|]+E[|y_d-y_s|]=\frac{M+N+2}{3} \quad 式(4.13)$$

假设$N_k$为网络中第$k$个节点的连通度数，同理，计算 mesh 网络中节点的平均度数$E[N_i]$为：

$$E[N_i]=\frac{1}{M\times N}\sum_{k=0}^{M\times N-1}N_k=4-\frac{2(M+N)}{M\times N} \quad 式(4.14)$$

可得，对于 mesh 网络计算过程的时间和空间复杂度分别为：

$$\begin{aligned}T&=(V^2+V\times E[N_i]+n_k\times E[d_{sd}]^2)\times a+(E[d_{sd}]+1)\times\beta\\&=[(M\times N)^2+M\times N\times(4-\frac{2(M+N)}{M\times N})]\times a\\&\quad+n_k\times(\frac{M+N+2}{3})^2\times a+(\frac{M+N+2}{3}+1)\times\beta\end{aligned} \quad 式(4.15)$$

$$\begin{aligned}S&=V+m\times n+2E+n_k\times E[d_{sd}]\\&=9M\times N-4(M+N)+n_k\times\frac{M+N+2}{3}\end{aligned} \quad 式(4.16)$$

同理，若未考虑链表中接收到的光路长度的大小，则对于 mesh 网络计算过程中的时间和空间复杂度分别为：

$$\begin{aligned}T_x&=[(M\times N)^2+M\times N\times(4-\frac{2(M+N)}{M\times N})]\times a\\&\quad+n_x\times(\frac{M+N+2}{3})^2\times a+(M+N)\times\beta\end{aligned} \quad 式(4.17)$$

$$S_x=9M\times N-4(M+N)+n_x\times\frac{M+N+2}{3} \quad 式(4.18)$$

因此，在 mesh 网络中限制链表接收的光路长度可以节省的时间和空间为：

$$T_x-T=(n_x-n_k)\times(\frac{M+N+2}{3})^2\times a+\frac{2(M+N)-5}{3}\times\beta \quad 式(4.19)$$

$$S_x-S=(n_x-n_k)\times\frac{M+N+2}{3} \quad 式(4.20)$$

其中，$n_x$是链表中在目的节点接收的未限制光路长度的光路数，间接地增加了链表接收光路的传播时间，而且目的节点会接收相对较多的光路数，因此相对地增加了存储光路的链表空间；由于链表中最长的光路长度变大，则其余光路在与其比较时将增加链路的匹配时间。

与此同时，将本章的多故障定位机制与多次使用单链路故障定位方法定位网络中出现的多条链路故障进行了比较。针对网络中出现的单链路故障情况，文献[15]中提出了 LVM 协议的故障定位方法，并将故障定位过程分为洪泛、暂停、多播、

匹配和总结五个阶段，其中，$D$表示网络直径的大小，$\alpha$为计算时间周期，$\beta$为邻居节点间的平均传播延迟，$E[d_{sd}]$为网络的平均路径长度，则该协议的时间和空间复杂度的计算公式分别为：

$$T_{LVM}=\sum_{i=1}^{5}T_i$$
$$=E[d_{sd}]\times\beta+D\times\beta+(E[d_{sd}]+2)\times\beta+E[d_{sd}]^2\times\alpha+\alpha \quad 式(4.21)$$
$$S_{LVM}=(m_l+n_l)\times E[d_{sd}] \quad 式(4.22)$$

在LVM协议进行故障定位的过程中，只需计算多播和匹配阶段的空间复杂度。在多播阶段，有限周边中的 sink 节点收到影响链路矢量 ALV 的副本，并将每条链路用$m_l$比特表示；在匹配阶段，$n_l$表示执行 sink 节点从$n_l$个 sink 节点接收带有 ALV 长度的匹配矢量。

本章提出的多故障定位机制，网络初始化和网元抽象阶段的时间为预配置时间，因此在定位网络故障的过程中只需考虑故障探测阶段和链路匹配阶段的时间，其时间复杂度的计算公式可以化简为：

$$T_{DM}=(E[d_{sd}]+1)\times\beta+n_k\times E[d_{sd}]^2\times\alpha \quad 式(4.23)$$

网络中若同时出现$n_f$个链路故障，利用本文方法一次就可以将这$n_f$个故障进行定位，假设使用 LVM 协议进行$n_f$次运算后也可以定位多条故障链路，由于在洪泛阶段网络产生故障告警，因此定位这些链路故障需要的时间复杂度为：

$$T_{n_f_LVM}=D\times\beta+n_f\times[E[d_{sd}]\times\beta+(E[d_{sd}]+2)\times\beta+E[d_{sd}]^2\times\alpha+\alpha] \quad 式(4.24)$$

由于网络中节点间的平均传播延迟远远大于计算时间周期，故使用本文故障定位机制可以相对节省的时间为：

$$T_{n_f_LVM}-T_{DM}=D\times\beta+(2n_f-1)(E[d_{sd}]+1)\times\beta$$
$$+[(n_f-n_k)\times E[d_{sd}]^2+n_f]\times\alpha \quad 式(4.25)$$

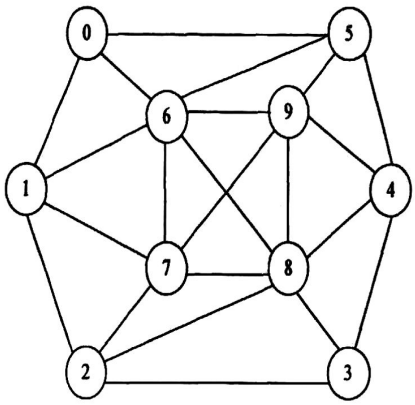
而本章故障定位机制需要为网络属性信息、队列和链表分配一定的存储空间，由于链表中存储了一定数量的光路，故占用的网络存储空间比 LVM 协议的存储空间相对较大，其节省的空间大小为：

$$S-S_{LVM}=(V+m\times n+2E+n_k\times E[d_{sd}])-(m_l+n_l)\times E[d_{sd}]$$
$$=V+m\times n+2E+[n_k-(m_l+n_l)]\times E[d_{sd}] \quad 式(4.26)$$

### 4.3.2 多故障定位率

根据本章提出的多链路故障定位机制，文中对 Small、COST239 网络的多故障定位概率进行了分析，首先根据多故障定位机制过程中的带有约束条件的网元抽象

方法对这两个网络进行网元抽象，并将网络中的所有节点映射到一个带有漏洞的 $m\times n$ 棋盘上。如图 4.5 所示，图 4.5(a)为 Small 网络拓扑结构，图 4.5(b)为将其进行网元抽象后，网络中的所有节点映射到 8×6 棋盘上的对应关系图，其中标示数字的方格代表网络中的节点在棋盘上的映射位置，棋盘上的灰色方格则为马步禁止访问位置，即棋盘的漏洞。



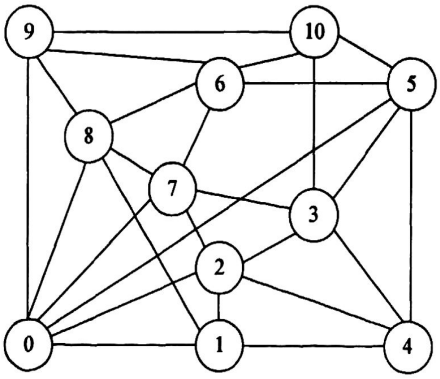

(a)Small 网络拓扑

| | | | | | |
|---|---|---|---|---|---|
| | 8 | 6 | 4 | | |
| 5 | 5 | 7 | | 8 | |
| 7 | 6 | 9 | | 8 | 9 |
| 4 | 9 | 6 | 8 | 7 | 7 |
| 9 | 9 | 8 | 6 | 7 | |
| 4 | | 8 | 6 | 9 | 1 |
| 4 | | 5 | 2 | | 1 |
| | 3 | 5 | 6 | 0 | |

(b)网络与棋盘的映射关系

图 4.5 Small 网络的网元抽象

图 4.6(a)为 COST239 网络的拓扑结构，图 4.6(b)是对网络进行网元抽象后，网络节点与带有漏洞的 10×14 棋盘方格之间的映射关系。其中网元抽象的复杂度与网络的节点个数及其节点的平均连通度有关，当网络的节点平均连通度较大时，每个节点在进行网元抽象的过程中需要占用更多的资源，而且需要多次对同一个节点进行多次寻找合法马步。




(a)COST239 网络拓扑

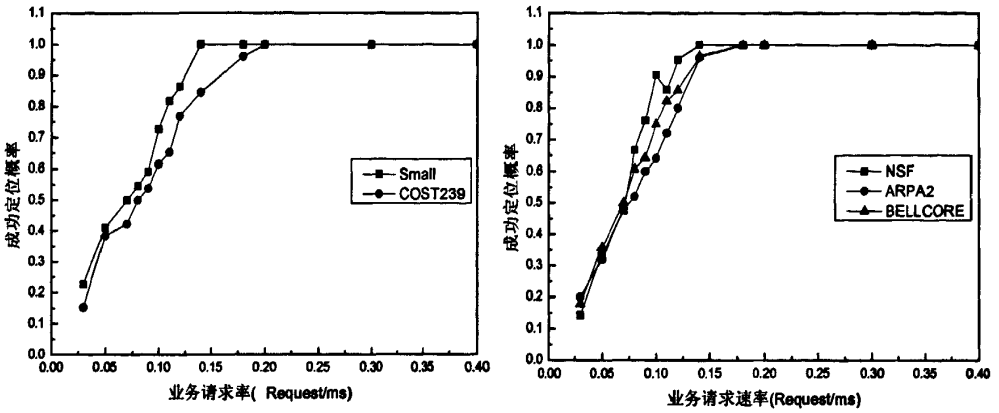
| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 0 | | | | 9 | 1 | | |
| | | 1 | | 0 | | | 0 | | 8 | | | | |
| | | | 1 | 1 | 9 | | 0 | | 0 | 2 | 6 | | |
| | 2 | | 8 | | | | | 7 | 7 | 9 | 5 | | |
| 3 | 6 | 4 | 0 | 0 | | | | 0 | 4 | 7 | | 8 | |
| | | 7 | 5 | | 5 | 2 | 3 | | 6 | 0 | | 3 | |
| | | 8 | 5 | 7 | | 0 | 5 | 2 | 2 | 4 | 8 | | 7 |
| 7 | 6 | 6 | 0 | | 3 | 3 | 1 | 5 | 0 | | 2 | | |
| 8 | | 3 | | 10 | 8 | 10 | 2 | 4 | 4 | | | 8 | |
| | 9 | 9 | | | | | | | | 1 | | | |

(b)网络与棋盘的映射关系

图 4.6 COST239 网络的网元抽象

故障定位概率为成功定位网络中的故障链路数占网络中总的故障链路数的比值。假设网络连接请求服从泊松分布，每个业务请求的探测信号在所有节点中服从均匀分布，每个连接的持续时间服从指数分布。假设开始定位前的故障链路总数为网络中的链路数，Small、COST239 网络故障定位概率的仿真结果如图 4.7(a)所示，同时

也对 mesh 网络中的 NSF、ARPA2、BELLCORE 网络的故障定位概率进行了仿真，其网络拓扑结构请参考图 3.5、图 3.7 和图 3.8，如图 4.7(b)所示为 NSF、ARPA2 和 BELLCORE 网络的多链路故障成功定位概率的仿真图。图 4.7 的仿真结果表明网络的故障定位概率随着业务请求速率的增加而显著增长，因为随着业务请求的增加，使网络中可用的光路数增多，目的节点将收到更多的有效光路，使得网络定位概率不断的增加，而当业务请求速率达到一定值时，虽然光路数目急剧增加，但有效的光路数并没有显著的变化，因此网络的定位率趋于稳定。




(a) Small、COST239 的故障定位率　　(b) NSF、ARPA2、BELLCORE 的故障定位率

图 4.7 多链路故障定位概率

## 4.4 本章小结

针对光网络中的多故障定位技术，现有故障管理文献中的研究相对较少。当网络发生故障时，若没有快速、有效的故障监测和定位技术，网络业务将很难得到迅速恢复，从而造成巨大的数据丢失和经济损失。利用骑士巡游理论，本章提出了一种光网络中的多链路故障定位机制，该机制将网络中的多链路故障定位过程分为初始化、网元抽象阶段、探测阶段、链路匹配阶段四个阶段，并对多故障定位过程的每个阶段进行了时间和空间复杂度分析，计算结果表明该定位机制能够相对降低故障定位的时间复杂度；同时对所提出方法的故障定位概率进行了计算机仿真，结果表明利用骑士巡游理论提出的多故障定位机制，能够在保证业务请求率的情况下达到对网络故障的完全定位。同时，文中对不同的网络拓扑进行了仿真，说明该多故障定位机制具有较好的适应性，从而为解决光网络中的多故障定位技术提供了一种有效的方法。

# 第五章 总结

## 5.1 论文小结

本文首先介绍了光网络的发展，光网络作为当今社会主要的通信网络，其承载的业务已经由语音业务扩展到数据业务以及多媒体为代表的宽带业务的方向发展，逐渐成为所有通信系统所依赖的主干通信系统。随着业务多样性的发展，光网络已经从由传统的点到点的业务传输方式向着光组网、智能控制等体现网络动态特性、可扩展性的方向迅速发展，同时随着人们对网络的依赖性越来越大，对网络性能以及对服务质量的要求也越来越高，因此在保障网络的通信质量、可靠性、生存性等网络性能方面，光网络技术需要持续不断地进行更新。

为了保障网络的通信质量、可靠性等网络性能，需要实时监测网络状态，并根据网络中监测设备发送的网络信息，及时采取有效的保护措施，维持网络的安全性和可靠性。关于故障管理的现有文献中均提出了故障监测和定位技术在网络生存性中的重要性，是故障恢复和保护的前提条件，并提出了相应的算法或机制对网络故障进行探测和定位，为了更好地利用网络资源，必须找出关于定位网络故障的有效算法或机制。由于网络性能监测为故障定位的判断提供可靠依据，因此在文中也对光性能监测技术及其面临的严峻形势进行了介绍和说明，同时在文中的 1.3.2 节和 2.2 节中对网络故障定位技术的研究现状进行了一定的分析，并给出了几种比较典型的光网络中的故障定位技术。

现有文献针对光网络中的单故障定位技术的研究相对较多，但是对光网络中的多故障定位技术的研究相对较少，虽然网络中同时发生多个故障的概率比较小，但是由于光网络业务传输的高速性，当网络发生故障时，若没有快速、有效的故障监测和定位技术，网络业务将很难得到迅速恢复，从而造成巨大的数据丢失和经济损失。而基于单故障定位的技术对于网络中的多故障并不一定能够做出快速、有效的故障判断，给后续的网络故障恢复或保护带来不确定性，因此针对网络中的多故障定位技术还需要更进一步的研究。

本文中将骑士巡游理论应用到光网络的故障管理中，并提出一种解决网络中单链路故障的定位策略和多链路故障的定位机制，并对 Small、NSF、COST239、ARPA2、BELLCORE 这五种网络的链路故障进行定位分析。

本文中的第三章给出了一种光网络中的单链路故障定位策略，由于网络拓扑、

无向图、棋盘三者之间存在一定的联系，因此，根据骑士巡游理论提出了使用棋盘对网络建立故障定位模型，然后提出一种网络节点度数感知的分裂方法将网络中的所有节点、链路一一映射到相应大小的$m\times n$棋盘上，并且能够反映网络中的节点与链路之间的连接关系；最后依据骑士巡游理论的思想，利用探测信号返回网络中的故障链路信息，通过查找静态映射表定位出故障链路的准确位置。同时对该故障定位策略的故障定位时间和链路覆盖长度进行了仿真和分析，对上面五种网络的仿真结果表明该策略的链路故障定位的平均时间均小于OSPF协议的故障定位时间40ms，链路覆盖长度相对小于圈覆盖算法的链路覆盖长度，因此基于骑士巡游的单链路故障定位策略能够有效的、快速的定位网络中出现的单个链路故障。

本文中的第四章提出了利用骑士巡游理论定位光网络中的多链路故障，该机制将网络中的多链路故障定位过程分为初始化、网元抽象阶段、探测阶段、链路匹配阶段四个阶段。根据骑士巡游理论，首先建立多故障链路定位的网络模型，利用网元抽象方法将网络中的所有节点映射到一个带有漏洞的$m\times n$棋盘上；然后在备选探测信号节点周期性发送探测信号信息，目的节点将收到的有效光路存储在链表中；最后通过链路匹配计算网络中的故障链路。文中对多故障定位过程的每个阶段进行了时间和空间复杂度分析，同时仿真了网络的故障定位概率，仿真结果表明利用骑士巡游理论提出的多故障定位机制，在保证业务请求率的情况下可以达到对网络故障的完全定位，从而为解决光网络中的多故障定位技术提供一种有效的方法。

## 5.2 未来工作

网络生存性问题一直备受关注，网络故障监测和定位作为保障网络生存性的关键技术之一，也不断的被研究着，然而实际网络运行过程中，故障发生的情况比较复杂，网络中链路故障发生的同时往往会引发节点故障等影响网络性能的问题，因此要对网络中引起链路故障的原因进行全面地分析，并给出准确有效地判断，从而尽量减小因故障所带来的网络性能恶化等问题。这将是以后网络故障定位研究中需要解决的一个问题，并寻找更加有效的故障定位方法。

文中将骑士巡游理论应用到光网络的故障管理中，并利用骑士巡游理论提出了解决光网络中的链路故障定位问题的一种方法，关于骑士巡游理论能否用到网络故障管理中的其它方面也在致力研究当中。例如将骑士巡游理论进行网络中的故障恢复等问题，利用本文中第四章的多故障定位的网络模型以及带有约束条件的网元抽象方法重构网络逻辑拓扑，当探测到网络故障时，在故障链路的上游节点处通过控制算法进行自动选路，避开故障链路，从而恢复网络数据的传输。

# 致 谢

在重庆这个美丽的城市，在拥有优美校园环境的重庆邮电大学中，我度过了三年的研究生生活。在这三年的校园生活中，学校培养了我对专业理论知识的学习能力和科研能力，更改变了我对生活的态度。

首先，我要感谢我敬爱的导师——王汝言教授，在这三年的研究生生活中，非常有幸得到王老师的谆谆教导，使我受益匪浅。在王老师的不断引导和鼓励下，逐渐培养了独立创新、解决问题的能力，以严谨的治学态度教导我们对科研任务要有责任心，要以求真务实、脚踏实地的精神对待自己的每一个研究任务。通过不断地交流与沟通，王老师不仅教给我们做学问的方法，更是教会我们许多做人的道理。非常感谢您在本文的选题、研究工作以及撰写过程中给予我的悉心指导和帮助，感谢您与我耐心的交流与沟通。在此，谨向王老师您致以衷心的感谢和崇高的敬意！

衷心的感谢吴大鹏老师您在我的研究工作和论文撰写过程中给予的帮助和耐心的指导，感谢您教给我的科研方法，在科研成果上给予的热心帮助，感谢您为我的研究课题付出的心血！

感谢光互联网与无线信息网络研究中心的刘焕淋教授、黄胜教授、向劲松教授，和张胜峰老师对我在三年的科研学习期间给予的指导和帮助！

感谢刘成耀、祖力、艾阳、代琦、王卓、左琼华、颜红以及实验室的其他同学在科研上和生活上给予的帮助，三年的朝夕相处，使我们同甘共苦，相互扶持，彼此理解的同窗之宜，彼此之间留下了不可磨灭的友情。感谢您们在生活和科研任务上给予的鼓励和提供的宝贵意见！

感谢所有默默关心、支持、帮助、批评和鼓励过我的人，你们的理解、意见和支持是我完成学业的重要保证！

最后感谢参加论文阅读和评审的各位专家，感谢答辩委员会的各位专家们在百忙之中参加我的毕业答辩和给予的指导，在此表示衷心的感谢！

# 参考文献

[1] 戈丽萍. WDM 全光通信网技术及其发展[J]. 内蒙古科技与经济. 2006, 22:102-105.

[2] 纪越峰等. 现代通信技术（第二版）[M]. 北京：北京邮电大学出版社. 2004, 273-276.

[3] 李维民，赵巧霞，康巧燕，黄海清. 全光通信网技术（第一版）[M]. 北京：北京邮电大学出版社. 2009, 118-135.

[4] Zhang Minglun, Zhang Yangan, Yuan Xueguang, Zhang Jinnan, Zhang Xiaoguang. Optical performance monitoring in reconfigurable OADM networks [C]. Lasers & Electro Optics & The Pacific Rim Conference on Lasers and Electro-Optics. 2009, 1-2.

[5] Chung Y.C.. Optical performance monitoring techniques: current status and future challenges [C]. 2008 34th European Conference on Optical Communication. 2008, 1-3.

[6] Islam M.S., Majumder S.P.. Optical and higher layer performance monitoring in photonic networks: progress and challenges [C]. 2009 11th International Conference on Advanced Communication Technology. 2009, 3:1591-1596.

[7] 景琦，赵峰，邓大鹏，解东宏. 透明光网络物理层中攻击、故障检测方法研究. 光通信技术[J]. 2005, 29(9): 14-16.

[8] Natu M., Sethi A.S.. Active Probing Approach for fault localization in computer networks [C]. 2006 4th IEEE/IFIP Workshop on End-to-End Techniques and Services. 2006, 25-33.

[9] 孙俊. WDM 全光网络的故障定位[J]. 光通信研究. 2002, 6:10-14.

[10] Stanic S., Subramaniam S., Sahin G., Choi H., H.-A.. Active monitoring and alarm management for fault localization in transparent all-optical networks [J]. IEEE Transactions on Network and Service Management. 2010, 7(2):118-131.

[11] Carmen Mas, Patrick Thiram. An efficient algorithm for locating soft and hard failures in WDM networks [J]. IEEE Journal on Selected Areas in Communications. 2000, 18(10):1900-1911.

[12] 刘兰英. 基于网络分割策略的 OBS 网络故障监测机制研究[D]. 重庆：重庆邮电大学. 2010, 10-12.

[13]Mas C., Tomakos I., Tonguz O. K.. Failure location algorithm for transparent optical network [J]. IEEE Journal on Selected Areas in Communications. 2005, 23(8):1508-1519.

[14]Ridha Rejeb, Mark S. Leeson, Roger J. Green. Fault and attack management in all-optical networks [J]. IEEE Communications Magazine. 2006, 44(11):79-86.

[15]A.V. Sichani, Hussein T. Mouftah. Limited-perimeter vector matching fault-localization protocol for transparent all-optical communication networks [J]. IET Communications. 2007, 1(3): 472-478.

[16]Mazen Khair, Burak. Kantarci, Jun Zheng, Hussein T. Mouftah. Optimization for minimizing fault localization time in all-optical networks [C]. 2008 International Conference on Transparent Optical Networks. 2008, 3:63-66.

[17]Mazen Khair, Burak Kantarci, Jun. Zheng, Hussein T. Mouftah. Performance optimization for fault localization in all-optical networks [C]. 2008 International Conference on Broadband Communications, Networks and Systems. 2008, 531-535.

[18]Mazen Khair, Burak Kantarci, Jun Zheng, Hussein T. Mouftah. Optimization for fault localization in all-optical networks [J]. IEEE Journal of Lightwave Technology. 2009, 27(21):4832-4840.

[19]Hongqing Zeng, Changcheng Huang. Fault detection and path performance monitoring in meshed all-optical networks [C]. IEEE Global Telecommunications Conference. 2004, 3:2014-2018.

[20]Hongqing Zeng, Changcheng Huang, Alex Vukovic. A novel fault detection and localization scheme for mesh all-optical networks based on monitoring-cycles [J]. Photonic Network Communications. 2006, 11:277-286.

[21]Bin Wu, Pin-Han Ho, Kwan L. Yeung. Monitoring trail: on fast link failure localization in all-optical WDM mesh networks [J]. Journal of Lightwave Technology. 2009, 27(18):4175-4185.

[22]Bin Wu, Pin-Han, Tapolcai J., Xiaohong Jiang. A novel framework of fast and unambiguous link failure localization via monitoring trail [C]. 2010 IEEE Conference on Computer Communications Workshops. 2010, 1-5.

[23]Tapolcai J., Ronyai L., Pin-Han Ho. Optimal solutions for single fault localization in two dimensional lattice networks [C]. 2010 Proceedings IEEE INFOCOM. 2010, 1-5.

[24]Ahuja S.S., Ramasubramanian S., Krunz M.. Single-link failure detection in all-optical networks using monitoring cycles and paths [J]. IEEE/ACM Transactions

on Networking. 2009, 17(4): 1080-1093.

[25] Yongning Tang, Al-Shaer E., Boutaba R.. Efficient fault diagnosis using incremental alarm correlation and active investigation for Internet and overlay networks [J]. IEEE Transactions on Network and Service Management. 2008, 5(1):36-49.

[26] Sheng Huang, Ming Xia, Charles U., Martel C., Mukherjee B.. A multistate multipath provisioning scheme for differentiated failures in telecom mesh networks [J]. Journal of Lightware Technology. 2010, 28(11):1585-1596.

[27] Kompella R.R., Yates J., Greenberg A., Snoeren A. C.. Fault localization via risk modeling [J].IEEE Transactions on Dependable and Secure Computing. 2010, 7(4):396-409.

[28] Babarczi, P., Tapolcai J., Pin-Han Ho, Bin Wu. SRLG failure localization in transparent optical mesh networks with monitoring trees and trails [J]. 2010 12th International Conference on Transparent Optical Networks. 2010, 1-4.

[29] Tapolcai J., Wu Bin, Ho P.-Han, Ronyai L. A novel approach for failure localization in all-optical mesh networks [J]. IEEE/ACM Transactions on Networking. 2010, pp(99):1-11.

[30] Bin Wu, Kwan L. Yeung. $M^2$-CYCLE: an optical layer algorithm for fast link failure detection in all-optical mesh networks [C]. 2006 IEEE Global Telecommunications Conference. 2006, 1-5.

[31] Jeffrey D., Gupta N., Gupta R.. Effective and efficient localization of multiple faults using value replacement [C]. 2009 IEEE International Conference on Software Maintenance. 2009, 221-230.

[32] Lee P.P.C., Misra V., Rubenstein D.. Toward optimal network fault correction in externally managed overlay network [J]. IEEE Transactions on Parallel and Distributed Systems. 2010, 21(3):354-366.

[33] Debroy V., Wong W.E., Xiaofeng Xu, Byoungju Choi. A grouping-based strategy to improve the effectiveness of fault localization techniques [C]. 2010 10th International Conference on Quality Software. 2010,13-32.

[34] Wotawa F.. Fault localization based on dynamic slicing and hitting-set computation [C]. 2010 10th International Conference on Quality Software. 2010, 161-170.

[35] Miao He, Junshan Zhang. Fault detection and localization in smart grid: a probabilistic dependence graph approach [C]. 2010 first IEEE International Conference on Smart Grid Communications. 2010, 43-48.

[36] Xiaoyuan Xie, Tsong Yueh Chen, Baowen Xu. Isolating suspiciousness from

spectrum-based fault localization techniques [C]. 2010 10th International Conference on Quality Software. 2010, 385-392.

[37] Steinert R., Gillblad D.. Towards distributed and adaptive detection and localization of network faults [C]. 2010 Sixth Advanced International Conference on Telecommunications. 2010, 384-389.

[38] .Yonggang Wen, Vincent W.S. Chan, Lizhong Zheng. Efficient fault-diagnosis algorithms for all-optical WDM networks with probabilistic link failures [J]. Journal of Lightwave Technology. 2005, 23(10): 3358-3371.

[39] Mazen Khair, Jun Zheng, Hussein T. Mouftah. Distributed fault localization protocol for multi-domain all-optical networks with partial power monitoring[C]. 2008 International Symposium on Computer and Communications. 2008, 788-793.

[40] Mazen Khair, Jun Zheng, Hussein T. Mouftah. Distributed multi-failure localization protocol for all-optical networks [C].2009 International Conference on Optical Network Design and Modeling. 2009, 1-6.

[41] 林新棋，吴大鹏. 全光传送网的生存性技术[J]. 现代电子技术. 2007, 30(19):49-51,54.

[42] Gordon, V. S. and T. J. Slocum. The knight's tour-evolutionary vs. depth-first search [C]. IEEE Congress on Evolutionary Computation. 2004, 1435-1440.

[43] 柏森，杨晓帆. 求马步图 Hamilton 圈的最优算法[J]. 计算机工程与科学，2000, 22(2):8-11.

[44] Huang Jian, Bai Sen. An efficient algorithm for the generalized (1,k)-knight's tours problem [C]. 2009 First International Workshop on Education Technology and Computer Science. 2009, 1:697-701.

[45] 张赞波. 无限大棋盘上马的遍历问题[J]. 广东轻工业职业技术学院学报. 2006, 5(3):15-17.

[46] 宁宣熙，宁安琪，吴薇薇，雷仲魁. 阻塞流理论及其应用（第二版）[M]. 北京：科学出版社. 2009, 4-200.

[47] 殷剑宏，吴开亚等. 图论及其算法[M]. 合肥：中国科技大学出版社. 2005, 1-6.

[48] Jiang Delei, Bai Sen, Dong Wenming. An ant colony optimization algorithm for knight's tour problem on the chessboard with holes [C]. 2009 First International Workshop on Education Technology and Computer Science. 2009, 1:292-296.

# 附录 攻读硕士学位期间从事的科研工作

1、项目研究

[1] 项目名称：下一代光交换网络故障容忍策略研究
项目类别：国家自然科学基金
项目编号：60972096

[2] 项目名称：面向多业务的光波带突发交换技术研究
项目类别：重庆市科委自然科学基金项目
项目编号：CSTC2009BB2285

2、论文发表

[1] 吴大鹏，李阳，王汝言. 基于骑士巡游的 Mesh 光网络链路故障定位策略[J]. 重庆邮电大学学报（自然科学版）. 2011, 23(1):1-5.

[2] Dapeng Wu, Yang Li, Ruyan Wang, Xin Cui. Knight's tour based fault localization mechanism in mesh optical communication network [J]. In the process of review.

3、获奖情况

[1] 艾阳，王卓，李阳. 光突发交换核心节点交换控制系统. 第七届中国研究生电子设计竞赛. 西南赛区团体三等奖. 2010.